Design Considerations in Aerospace Multilevel Structural Optimisation

By

Daniel Aloysius Francis

Department of Aerospace Engineering
University of Bristol

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Doctor of Philosophy in the Faculty of Engineering.

December 17, 2019

31,750 words

ABSTRACT

Multilevel optimisation (MLO) refers to increasing the level of detail present in the optimisation problem through a decomposed set of subproblems coordinated though some governing strategy. Within engineering MLO can be used to improve the preliminary structural design process by accounting for additional detailed constraints and refined levels of fidelity typically not represented during conceptual sizing. Increasing the level of design fidelity earlier allows for a greater range of nuanced solutions, exploitation of structural reserves, improved robustness and greater accuracy in fundamental design decisions.

This research focused on practical considerations of MLO, applying algorithm from the major subsets of research identified within the literature to an industry guided design problem based on a half-span metallic wing-box design. This involved developing a generic modular framework to explore integrating detailed structural analyses for sub-components within a global wing sizing phase to minimise structural mass. Each algorithm was tested using varying assumptions and models, gradually increasing the difficulty of the design problem. A range of indices were identified to best assess and select the appropriate methods based on quantitative problem profiling.

Monolithic methods were not robust enough for the most challenging problems but produced viable design solutions orders of magnitude more efficient than other methods tested. Dual distributed methods were the most reliable but overlapping algorithms, the breadth of techniques, tuning parameters and partitioning considerations means it can be challenging to select the appropriate algorithm, tune state parameters and produce useful results. Of the optimisers tested the most robust was a dual distributed method, augmented analytical target cascading with gradient-based partitioning. This method performed as good as or better than the baseline monolithic single-phase algorithm under most test conditions.

Both monolithic and distributed MLO algorithm led to mass reductions of between 4-7% compared to traditional single-level design. Practical considerations for the application of MLO methods in industry include measuring the meaningfulness of decomposition to ensure subproblems capture all the information of the original non-parsed problem. Additionally, pessimism can be used as a measure of competition or adversity between the system-subsystem problems, quantifiable through a range of factors which can significantly impact the applicability of any given method. And finally solution space screening must be used to isolate non-convex constraints and properly scale dissimilar design responses into appropriate subproblems.

Dedication and acknowledgements

I dedicate this dissertation to my family and many friends, too many of whom to thank individually but without all of there continued support and encouragement this work would not have been possible. With special mention to the staff and students of 2.80, the AWI project and Manor Hall. I am grateful to my siblings, my parents - Jacqueline and Anthony Francis, who have provided me moral and emotional support throughout.

I would also like to acknowledge the help of my supervisors and supporting academics for there help and insight, this includes but is not limited to: Dr Olivia Stodieck, Dr Alberto Pirrera and Professor Jonathan Cooper. I wish to thank my reviewers who were more than generous with their expertise and precious time; Professor Vassili Toropov and Dr Luiz Kawashita.

This research was funded by the UK Aerospace Technology Institute-Agile Wing Integration (AWI) project (TSB-113041). J.E. Cooper holds a Royal Academy of Engineering Airbus Sir George White Chair in Aerospace Engineering and A. Pirrera EPSRC Early Career Research Fellowship.

> *“It is idle, having planted an acorn in the morning, to expect that afternoon to sit in the shade of the oak.”*
>
> Antoine de Saint-Exupéry, *Wind, Sand and Stars*, 1939

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programs and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

TABLE OF CONTENTS

List of Algorithms

List of Tables

LIST OF FIGURES

FIGURE **Page**

NOMENCLATURE

Abbreviations & Acronyms

AAO All At Once

ADMM Alternating Direction Method of Multipliers

ATC Analytical Target Cascading

BLISS Bi-level Integrated System Synthesis

BLPP Bi-level Programming Problems

CFD Computational Fluid Dynamics

CO Collaborative Optimisation

CPOS Coupled Post-Optimality Sensitivity

DFEM Detailed Finite Element Model

DSP Decision-Support Problems

ECO Enhanced Collaborative Optimisation

EGO Efficient Global Optimisation

FE Finite Element

GA Genetic Algorithm

GFEM General Finite Element Model

KKT Karush-Kuhn-Tucker

KS Kreisselmeier-Steinhauser

LHS Latin Hyper-cube Sampling

MAC Modal Assurance Criterion

MaxC Max Constraint Violation

MDO Multidisciplinary Optimisation

MHP Man-Hole Plank

MLO Multilevel Optimisation

PDF Probability Density Function

PSO Particle Swarm Optimisation

RBE Rigid Body Element

RSME Root Mean Squared Errors

SE Super-element

SEQ Sequential Level Mixing

SQP Sequential Quadratic Programming

XDSM Extended Design Structure Matrix

Properties

ν Poisson's ratio

ρ Density

E Young's modulus

Symbols

$\boldsymbol{v}$ Design variable vector

$\boldsymbol{x}$ System-level design variable vector

$\boldsymbol{y}$ Subsystem-level design variable vector

$\boldsymbol{z}$ Slack variable design vector

δ Displacement

ϵ Design tolerance

λ Buckling eigenvalue

μ Lagrangian multiplier

ϕ	Penalty function
ρ_c	Nominal 'draw-down' aggregation factor
σ	von Mises stress
c	Inequality constraint
C_d	Design capability
D_f	Design freedom
F, f	Objective function
h	Equality constraint
I_C	Information certainty
K	Super-element stiffness matrices
KS'_d	Aggregation sensitivity tolerance
L_p	Augmented lagrangian objective function
M	Objective mass
N	Total Number of subsystems
P	Super-element displacement matrices
PR	Performance range
q	Consistency constraints
S	Selection matrices
t	Skin thickness
w	Penalty violation weighting

Subscript

0	Linking variable
i	Subsystem variable number
m, p	Number of control points
N	Total number of subsystems
rf	Normalised reserve factor

Superscript

(0)	Initial conditions
U, L	Upper and lower bounds
k	Current step number
$*$	Optimal value

Accents

$\tilde{x} \rightarrow x$	Surrogate value of variable
$\hat{x} \rightarrow x$	Independent copy of variable
$\bar{x} \rightarrow x$	Aggregated single value from a vector

GLOSSARY

Active constraint · An active constraint is any factor which is causing the limitation on the objective function, those which are the closest to being infeasible. A non-active constraint is one which is eminently feasible, and is not limiting the objective function.

Acyclic · Not displaying or forming part of a cycle.

Alternating algorithms · Iterate between solving a system-level coordination problem and sub-problems in sequence.

Cooperative behaviour model · A function which describes the interaction of two or more persons or organisations directed toward a common goal which is mutually beneficial.

Decision-support problem · Is an optimisation problem in which multiple sub-problems must coordinate a distributed set of decisions, with support from one another.

Design Decomposition · Breaking a design down into simpler bite-sized problems one can feasibly resolve a greater number of localised problems.

Design space · Refers to the range or variation in input design variables for a function or design model.

Distributed method · An optimisation architecture where a single design problem is solved iteratively as a collection of sub-problems and a global coordination problem.

Dual decomposition · Allowing for multiple sub-problems to independently vary local copies of linking variables, with relaxed associated coupling constraints.

Equivalence of copies · Requirement in dual optimisation problems to ensure duplicate design variables across multiple models are equal within some tolerance in the final solution.

Finite differencing · A method of obtaining the differential of a function, obtained by incrementing dependent variables of a function by a fixed amount.

Finite element method · Within structural design and engineering this refers to a numerical method which discretises a geometry into an idealised mesh to be solved analytically.

Gradient-based method · Any deterministic approach within optimisation where the search direction is defined using the gradient of the function at the current design point.

Hierarchical systems design · Refers to systems layout that is usually organised into the shape of a pyramid, with each row of objects linked to components and parts directly beneath it.

Linking variables · Shared variables and inputs which exist exist in multiple sub-problems within a distributed or hierarchal system.

Monolithic · An optimisation architecture where a single optimisation problem is solved in a defined sequence.

Nested algorithms · Decomposed sub-problems exist within some outer system or coordination problem. With each design-step of the outer problem a set of sub-problems our solved.

Optimiser · Refers to any software capable of solving a constrained or unconstrained multivariate optimisation problem.

Optimistic design problem · Assumes a lower level sub-problem selects the solution which also leads to the best outcome for the upper level problem.

Pessimistic design problem · Assumes a lower level sub-problems solution has selected the worst-case result for the upper level problem.

Portability · The ability for an object or method to be transferred from one system to another.

Primal decomposition · Maintaining the original form of the complete design problem, with consistent linking variables between multiple sub-problems.

Principal agency · A property of any entity capable of making independent decisions.

Problem Partition · Grouping of design variables and responses into a set of sub-problems, with the assumption that each can be reasonably decoupled from one another.

Quasi-separable function · Mutli-model design systems which are only linked through shared input variables with no associated multi-system dependent objectives or constraints.

Relaxed feasibility · Distributed optimisation architectures which allow for open design constraints. Usually some ramp or prohibitive penalty functions ensure final design feasibility.

Solution space · Refers to the range or variation in responses for a functions or design model.

Stackelberg competition · A strategic game in economics in which the leader moves first and then the follower move sequentially, usually the pair compete for some quantity.

Stochastic method · Any approach which introduces random elements into its methodology in-order to generalise a deterministic method. The result being the same set of parameter values and initial conditions may converge to a range of different outputs.

Strict feasibility · Distributed optimisation architectures which require that all design constraints are satisfied, and optimality conditions are met for each iteration.

Surrogate/ Meta-Model · A representative function which emulates the behaviour of some true function, whilst being computationally cheaper to analyse.

Target propagation · Used to associate function objectives across a multi-model system with one another.

CHAPTER 0

INTRODUCTION

0.1 The problem

Aircraft structures are often designed with competing interests and requirements. They must be strong enough to withstand all loads experienced during an aircraft's lifetime, maintain design and shape integrity, safely accommodate a payload and support a range of ancillary systems. Simultaneously, they must be as light as possible to achieve optimal performance, reduce environmental impact through lower fuel burn or advanced production methods, be safe and manufacture feasible and overall fiscally efficient. To produce the most effective design solution and satisfy these requirements optimisation is used to automate portions of the design process, a mathematical method of selecting the best design relative to a given set of feasible alternatives. Optimisation research has become a wide and varied topic of discussion within the literature, however, techniques have yet to see much adoption as a design tool within industrial design engineering outside of the operational sector, networking or computer sciences [1].

Before one can use optimisation to aid in design we must redefine the design problem as an optimisation problem with the following terms [2]:

- The objective function; a the real-valued function whose value is to be either minimised or maximised over the set of feasible alternatives. This can be a simulation-based analytical model of the design, measuring some quantity to determine as the final goal which the algorithm is attempting to minimise, often mass, fuel consumption or cost.

- Design Variables; those variables whose values can vary over the feasible set of alternatives in order to either increase or decrease the value of the objective function. These describe the system based on several degrees of freedom, defining the overall system characteristics, such as: geometry, loading, material selection, etc.

- Constraints; performance metrics the final solution must not exceed in-order to be feasible. These may describe some upper design loads, component failure or volume requirement, with the result being some limitation outlining a feasible region of solutions.

Engineering design uses a range of specialist software and analytical models and integrating them into the optimisation problem can create incredibly useful design tools. Within an analytical model one may use some Finite Element (FE) analysis, Computational Fluid Dynamics (CFD) or other methods capable of simulating the 'real' system behaviour with sufficient fidelity[1]. In general, engineering design optimisation is broadly dependent on design evaluation effort (*cost* of analyses) and degrees of freedom (*number of design variables*). Mixing discrete and continuous design variables and simulation non-linearity generally increases the mathematical complexity of the problem further [1].

Interpreting a design problem into an optimisation problem can be difficult and can lead to costly errors if done poorly. Designing the optimisation problem, evaluating the results and implementing them within real-world designs still requires a significant amount of engineering judgement. Meanwhile the growth in computing power has meant that engineers have been able to pose more significant problems for a range of mechanical structures; using computer-aided engineering tools of increasing fidelity. Increasing the fidelity and complexity of models improves the realism within computer simulations, the accuracy of results and improves the overall design through numerical and efficient exploitation of a more comprehensive design space.

Once an optimisation problem has been set up there is no guarantee that, based on the interpretation of the design, the problem itself has a feasible set of coherent optimal solutions. Before an optimisation problem can be solved there are some essential criteria to assure the convergence to some optimum design, for example, variables and system responses must be of similar orders of magnitude, no set of variables can either dominate or have little to no effect on the system responses, variables must be separable without interference with one another. Following these conventions ensures the optimisation problem is numerically determinate. [3] .

The size and expense of the problem are scaled depending on when and how the information is used in the design process. For complex structures, there is still a trade-off between using simple models with faster evaluations of limited accuracy across a more extensive design space against the use of more realistic but expensive models within a smaller design space [4]. Global approximations at the systems design stage allow for quick and broad studies of multiple concepts or novel solutions. These top-level decisions often lack detail and use a range of limiting assumptions, especially when the level of expertise required for making robust design decisions are not available.

[1] In this work, 'fidelity' refers to the degree of exactness with which one can replicate and describe the physical parameters of a system.

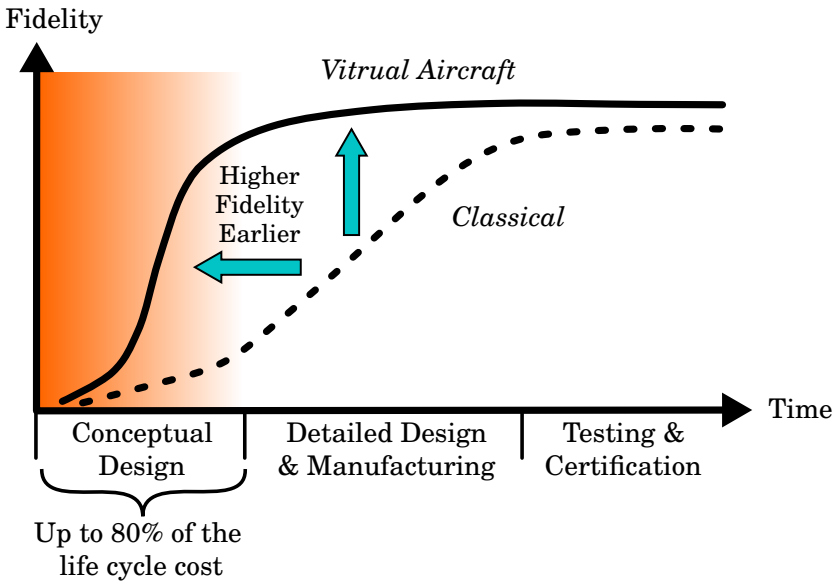


FIGURE 0.1. Solution Fidelity in Product Development; Cost influence of increasing model fidelity earlier in the design process [5].

Approximately 80% of life cycles cost for an aircraft are imposed in the design decision made during the concept phase [5]. Increasing fidelity earlier in the product development using a virtual recreation of the aircraft with as much information as possible, such as in figure 0.1, can mitigate potentially costly missteps later in the design cycle.

In figure 0.1 the 'classical' approach to product development has design fidelity increasing gradually throughout, often linked with an increase in associated personnel and specialist teams or expertise. Using the 'virtual aircraft' approach, with a robust set of tools to fully express the aircraft in high fidelity, one can make more accurate design decisions earlier in the product life cycle, ideally when the associated risk and cost is lowest. [5].

It is difficult during the concept-phase of structural design to ensure there is a level of fidelity to account for detailed constraints accurately alongside system-level design decisions. Designs created using simpler approximate models often lead to overly conservative designs, overly optimistic under design or those without sufficient detail to support further design. These problems are magnified when applied to novel or unconventional configurations without previous experience to refer to [6].

0.2 The Solution

Multilevel Optimisation (MLO) refers to the *decomposition* of a system model into a distributed set of isolated models [7]. This approach is simply a method of design partitioning into sets of variables, or constraints within separated functions, as shown in figure 0.2.

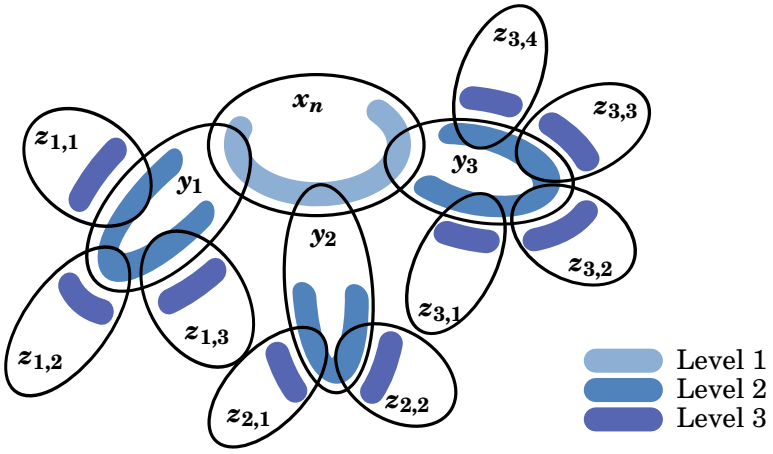


FIGURE 0.2. Venn diagram for a three-level system partition of design variables the design variable vector, $\boldsymbol{v}$ has been partitioned into subsystems $[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}]$ [7].

By breaking a design down into more straightforward bite-sized problems, one can feasibly resolve a more substantial problem through many coordinated localised problems and this is a process known as *partitioning*. The MLO form of the problem is partitioned into identifiable groups of objectives, variables and constraints, which form a hierarchic collection of, ideally, weakly coupled subproblems [8].

Design decomposition is the process of modelling a system through a series of smaller components, parts or substructures. Problem partitioning is the process of splitting a collection of design variables and responses into a phased or distributed algorithm in order to produce a more practical solvable problem.

Structural design routinely involves decomposition, breaking down the whole problem into a defined set of sub-analyses [9].

MLO expands the design space with more detailed and nuanced information, which increases the confidence in early-design decisions. Moreover, MLO allows the exploitation of design margins reserved for when conservative global approximations and simplified assumptions used during the initial design phase. The resulting optimisation problem is often large, featuring a range of complex non-trivial interactions and sets of numerically indeterminate design variables.

Decomposition of the overall design allows for specialist teams, tools or solvers; the result is a more considerable amount of detailed information available to the designer. Partitioning the optimisation problem into more manageable sub-problems allows for greater confidence in the optimisation algorithm and the efficient distribution of work. The downside being that although the problem is separable to a degree the design domain is still limited to a single design discipline, with many design variables cross the boundaries of a individual subproblem. The result is a greater need for coordination of the decomposed elements to produce a coherent overall system-level feasible solution [10].

0.3 Rationale and Novelty of Thesis

This thesis investigates methods for structural sizing, focusing on exploring the integration of detailed structural analysis for sub-components into the global wing sizing phase. The resulting multilevel sizing capability for wings must ensure fast preliminary sizing of the entire component while capturing an increased level of detail at the sub-component level.

Solving an MLO problem is not trivial, with the resulting optimisation problems being much larger and difficult to solve. Luckily there are many different architectures and formulations of optimiser[2] designed to solve the MLO design problem. In recent years the growth in computing power and increased accuracy of modelling has renewed interest in distributed multi-model design optimisation for complex structures. Increasing the fidelity present in structural modelling during the conceptual sizing phase allows for exploitation of structural reserves in derivative wing concepts. Similarly including refined analyses and additional material typically not presented in idealised structural models opens the design space to new potential weight savings. Additional material may include the masses of additional repeating substructures or non-generic components not accounted for in the early design process.

Due to the range of applications and potential for design improvements there is a breadth of research in solving the MLO problem. However, even with a range of optimisers to choose from, there is often little supporting research for the appropriate selection, tailoring of algorithms and comparison of solutions for large-scale MLO problems. Optimisers are usually specialised for fundamentally different systems. These potentially include costly high fidelity models where the optimisation problem is designed to be as computationally efficient as possible; problems designed for mixed discrete and continuous variables; or non-differentiable analytical models. In MLO, there is very little numerical bench-marking or justification of solvers and how they solve specific types of problem. Most surveys focus on qualitative evaluations of a range of solvers [12–15].

[2] In this work, term ‘optimiser’ refers to any software capable of solving a constrained or unconstrained multivariate optimisation problem [11].

The novelty in this PhD is the investigation into MLO research, assess a range of viable of optimisers and finally to present a range of quantitative measures to evaluate the suitability optimisation methods. This would include the classification and selection criteria to select an appropriate optimiser based on the design problem. The other investigative objectives presented herein include:

i Development of a generic modular framework of aircraft wing design problems; offering a range of possible geometries and an expandable portfolio of detailed part-level and non-generic substructures based on current best practise.

ii Investigate and implement a range of MLO architectures based on the major subsets of research identified as part of a literature review, including monolithic single-phase optimisers, and surrogate-based primal distributed optimiser and augmented gradient-based dual distributed optimiser.

iii Two different design cases were assessed using a tip loaded and metallic cantilever rectangular wing-box model. The optimisation problem objective was to minimise the structural mass subject to stress, deflection and buckling constraints. The first test case was an optimistic design problem[3], where the majority of active constraints were in the system-level analyses. These types of problem are best suited for monolithic optimisation algorithms, concatenating the entire design problem into a single-phase of simultaneous analysis and design. The second design case was a pessimistic[4] design problem, which shifted the active constraints to subsystem-level analyses, where distributed optimisation algorithms performed best.

iv The most successful algorithms were then tested on a representative wing-box model with respect to an aeroelastic load case, better replicating the actual design problems within the concept-phase in an industrial design environment.

[3] Creates a favourable system:sub-problem relationship which assumes a lower level sub-problem selects a solution which also leads to the best outcome for the upper level problem.

[4] Creates a competitive system:sub-problem relationship which assumes a lower level sub-problems solution has selected the worst-case result for the upper level problem.

0.4 Thesis Outline

The thesis consists of 7 chapters. Chapter 1, provided an introduction and background to the investigation, summarising the aims and contributions of this work.

Chapter 1: *Background & Literature Review*;
The literature review provides an introduction to the major topics within multilevel optimisation. The review will detail the significant subsets of MLO research, classes of optimisation architecture and the current state-of-the-art practises. This review ends with a summary of key findings and a discussion of the research within this thesis, scope of the work and how it fits into the narrative of MLO research.

Chapter 2: *Modelling & Design Parameterisation*;
Covers the development of models, mirroring the current state of design and level fidelity used in an industrial context. Further discussion in this chapter details the parameterisation, decomposition and implementation considerations for engineering problems at multiple levels of design.

Chapters 3, 4 & 5: *Multilevel Optimisation Techniques*;
These chapters present MLO architectures based on the main subsets of MLO. Each optimisation procedure is outlined with a problem statement, an exploration of methodology and a schematic procedure for integration. The results of each optimisation are assessed in-turn, with trade studies for state-parameters.

Chapter 6: *Optimisation Comparison & Discussion*;
Directly compares the results from each optimisation technique, with additional consideration for the optimality and assumptions made in each and meaningful metrics to test the quality and applicability of each method. Successful algorithms were then tested for an additional wing-box model.

Chapter 7: *Final Conclusions & Future Work*.

CHAPTER 1

BACKGROUND & LITERATURE REVIEW

Optimisation and coordination of multilevel algorithms covers a broad and diverse field of research. Both in and out of the field of engineering design, a range of strategies have been presented for an array of potential problem types. This chapter provides an appropriate background understanding of the topics as well as a literature review of the current state of research; exploring terms, methods, and conventions used to provide a baseline for further discussion. This chapter ends with a summary of the current state of the art methodology of key findings and themes presented within the literature.

Multilevel optimisation (MLO) is simply about increasing the level of detail present in the optimisation problem. In practise MLO and decomposition refers to the partition of system models into a distributed hierarchical architecture with each resulting subproblem solved within reasonably separable, isolated functions. Within structural design, MLO allows the incorporation of detailed and more realistic subsystem models earlier in the design process. Improved, detailed information available earlier in the design process allows for more robust decision making, considering a more substantial range of constraints, and exploiting weight savings within a more nuanced design space.

In-order to better discuss MLO, a baseline understanding of optimisation theory, terms and algorithms are necessary and are present in the next chapter.

1.1 Defining the Optimisation Problem

Having discussed the role of optimisation and how to define an optimisation problem, we must also address the optimisation algorithm attempting to solve it. An optimisation algorithm is one that attempts to minimise some objective function $F(\boldsymbol{v})$ for a decision variable vector, $\boldsymbol{v}$, usually subject to some limiting inequality constraint function, $c(\boldsymbol{v}) < 0$, an equality constraint $h(\boldsymbol{v}) = 0$. The problem presented is bounded in the real domain with limits $\boldsymbol{v}^{\mathrm{L}} < \boldsymbol{v} < \boldsymbol{v}^{\mathrm{U}}$, defining the search space. An example of such a bound-constrained optimisation is stated in equation (1.1).

$$\begin{aligned} \underset{v \in IR}{\text{minimise}} \quad & f(\boldsymbol{v}) \\ \text{subject to} \quad & c(\boldsymbol{v}) < 0 \\ & h(\boldsymbol{v}) = 0 \\ & \boldsymbol{v}^{\mathrm{L}} < \boldsymbol{v} < \boldsymbol{v}^{\mathrm{U}} \end{aligned} \tag{1.1}$$

A minimum design solution, $\boldsymbol{v}^*$, is when no combination of decision variables exist in the solution space where the objective function $F(\boldsymbol{v}) < F(\boldsymbol{v}^*)$. Generally the conditions to test whether an algorithm has found the optimum is to test either:

(a) The Karush-Kuhn-Tucker (KKT) conditions have been satisfied [3].

(b) The change in decision variable is less than the minimum step size for an appreciable change in system response (gradient, $\nabla f(v) \rightarrow 0$).

(c) The objectives and constraints have been met within some given tolerance, within a steady solution.

There are many different types of optimisation algorithms, but the most direct method is always using a gradient-based approach. This approach examines the variation in the solution space (a scalar field of possible responses to the given model), to find a feasible and optimal point

in the design space (a vector filed of possible inputs to the given model). Algorithms directly use the gradient of the scalar field of a models solution space to find an optimum position. However, when gradient governs search direction, it is possible to fall into locally minimum solutions, when better solutions may exist "over the hill". Hence why sometimes gradient-based methods are referred to interchangeably as local optimisers.

If a single locally optimal solution exists within a solution space as its true global optimum, a function is said to be *convex*.

Implicit definitions of convexity include:

(a) For any pair of feasible solutions, the average of those two solutions is also feasible. The solution of the average must also be as good or equal to those of the initial pair.

(b) At no point is there a change in sign of the second differential of the objective or constraints functions.

(c) If a system is convex, by definition a locally optimal solution to f must also be the global optimum.

All optimisation problems are either convex or non-convex. Non-Convex functions have multiple, distinct, locally optimal solutions. Most analytic models, especially those based on real systems (such as CFD or FE), are assumed non-convex unless proved otherwise. Discrete and integer variables or any discontinuities in the design space will also result in a non-convex function [2].

When selecting an optimisation algorithm, one must consider the problem formulation: objective function, types of variables, number of objectives and constraints. Optimisation algorithms generally fall into two classes, gradient algorithms and stochastic algorithms. The later of the two sometimes referred to as global optimisers. Gradient-based are best for convex systems, while stochastic methods are better for non-convex or indeterminate systems both are widely used in the literature and industry. The differences between the two classes of optimiser can be great, some of the more illustrative examples and those used during this work have been reviewed below.

1.1.1 Stochastic Optimisation Methods

Stochastic methods tend to be based on real-life processes or introduce random elements into the algorithm. The result is that stochastic algorithm may not converge to the same solution every time, but generally yield good results for a broad range of problems. Most stochastic methods only evaluate cost, with limited constraint handling, as such minimal assumptions are needed on the models, solution space or design space. They are derivative-free globally optimal solvers making them ideal for systems with non-linear multi-modal objective functions or mixed continuous/discrete variables.

However, stochastic algorithms are often population-based or involve investigating significant portions of the design space, which becomes inefficient for high dimensional problems. For large-scale optimisation problems, these algorithms become prohibitively computationally expensive [16].

There are a wide range of stochastic optimisers, but the following are most widely used methods within the literature:

Genetic Algorithms (GA): Are a form of evolutionary algorithm, initialised with a random population within a discrete design space, each generation the best results go through a selection process. The weakest designs are discarded. The best (or fittest) solutions are combined to repopulate before the next generation, applying a random mutation factor to each new solution, improving the mean population quality with each subsequent generation.

GA have become a mainstay in optimisation across a range of fields, especially within engineering optimisation and structural design. Their wide-spread usage is mostly due to the limited information and assumptions needed about models; they can be readily applied to functions with uncertain or unconventional design spaces [1]. However, GA cannot guarantee optimality due to the nature of introducing random mutation and distribution within the design space. Although they can locate the region in which the global optimum exists, they can take a long time to locate an exact solution within said region [17].

Particle Swarm Optimisation (PSO): A number of particles are moved around the design space, each represents a solution s^k to the optimisation, given in equation (1.2), where α is the current solution, β is the weighted acceleration towards the individual particles best solution $\left(v^k_{(i)}\right)$ and γ is the weighted acceleration towards the swarms' best solution $\left(v^*\right)$. Each particle is randomly accelerated towards good solutions found so far and continually adjusts its speed and trajectory in the search space based on this information, moving closer towards the global optimum with each iteration. As seen in nature, this computational swarm displays a remarkable level of coherence and coordination despite the simplicity of its individual particles [18].

$$s^k = \alpha s^k \pm \beta\left(arg \min_{v \in x^k_1 \cdots x_n} f(v^k) - v^k_{(i)}\right) + \gamma\left(arg \min_{v \in x^k_1 \cdots x_n} f(v^k) - v^*\right) \tag{1.2}$$

As long as the derived stability conditions and the correct combination of tuning parameters can be found, PSO can accurately find the global optima of most search spaces in fewer iterations than a traditional GA for a structural design search space [19].

1.1.2 Gradient-based Optimisation Methods

Often known as a local or deterministic method, the search direction is defined by the gradient of the function at the current point. Gradient-based optimisers are the most effective for highly-dimensional design problems and are capable of handling thousands of designs variables. However they are only capable of finding the local minima to the given starting condition for any single solution.

Gradient-based methods generally fall into several broad categories, with variations and expansions depending on whether applied to a constrained or unconstrained problem [20]:

Steepest Descent: Find the greatest negative gradient at any given point, moving a step size , δv, in that direction. This can be used to find a partial solution but requires there to be a minimum point $\nabla f(v)$, meaning it can only be used on uni-variate bounded solutions [3].

Newton's Method: Or the Newton-Raphson method is a re-arrangement of the Taylor series approximation, written in matrix form where step size $\nabla f(v+\delta v) \approx \nabla^2 f(v)\delta v$. This is a much more aggressive than steepest descent in that it takes larger step changes which may lead to divergence within narrow design spaces and requires you to solve for the Hessian. However, it is possible to find the solution to a wider range of problems and can be modified to improve stability [3].

Conjugate Gradient: Is the most prominent iterative method for solving sparse systems of linear equations. This method modifies the steepest descent method to reduce the number of steps by effectively choosing the next step to set the gradient to zero. In practice at each iteration k, step direction d^k is given by Equation (1.3) using the method of conjugate directions. The search direction is constructed by conjugation of the residuals, not increasing components of gradient, $\nabla f(v)$ that have already been nulled. This method requires less computation than the damped/ backtracking variants of the Newton-Raphson method [21].

$$d^k == \begin{cases} -\nabla f(v^k) \\ -\nabla f(v^k) + \left(\frac{\nabla f(v^k).\nabla f(v^k)}{\nabla f(v^{k-1}).\nabla f(v^{k-1})}\right).d^{k-1} \end{cases} \tag{1.3}$$

Unlike many other gradient-based methods, conjugate gradient methods can be successfully applied to large scale non-linear Multidisciplinary optimisation (MDO) problems with a high degree of efficiency and success when compared to stochastic optimisers like GA [16]. When coupled with analytic adjoint methods for evaluating derivatives (linearizing the governing equations using residual variables) the computational cost of evaluation is independent of the number of variables. An ideal solution for large-scale optimisation problems [22] and has been shown to work with high-fidelity aircraft MDO [23].

Sequential Quadratic Programming (SQP): Transform the constrained problem into a Lagrange subproblem to be solved iteratively. The objective function is represented by a quadratic approximation of the real system with linear approximations of constraints. The Lagrange multipliers account for deviations in magnitude and sensitivities of objective and constraints, as well as accounting for non-active constraints[1]. The representative Lagrange form of the functions is then solved in place of the original, reassessing the new potential optima and then iteratively updating multipliers and the accuracy of approximations.

If the solution space is convex, then the KKT conditions are both necessary and enough for global optimality. SQP based methods attempt to solve for KKT equations by computing Lagrange multipliers directly, guaranteeing super-linear convergence by accumulating second-order information regarding the KKT equations using quasi-Newton procedures. SQP outperforms most methods in benchmark studies in terms of efficiency, accuracy and successful solutions [24, 25].

1.1.3 Hybrid Algorithms

When the gradient alone is used to direct the search for an optimum, it will always converge to a local solution, meaning the starting position for algorithms must be reasonably close to the global solution. Meanwhile for larger functions, stochastic methods can become prohibitively inefficient depending on the number of variables. Despite the many recent advances in both gradient and stochastic methods, hybrid algorithms are still the most efficient solution for highly-dimensional non-convex design problems.

For many realistic problems, optimisation algorithms and approximate modelling methods are nested with one another to improve efficiency and overcome the deficiencies present in a single optimiser.

Stochastic Global, Gradient-based Local: Integrating features of stochastic optimisers for global search, and local gradient optimisers in the latter portion to improve convergence and efficiency [1, 26, 27]. For large-scale problems GA is often combined with either: (a) gradient-based optimisers, where a GA finds the general location of optima within a search space and then trades-off to a local model for fine tuning of variables [28, 29], or (b) approximate models to improve accuracy, where surrogate models are used in order to speed up evaluation of the wider design space [30].

Surrogate Assisted Optimisation: Surrogate models are used in place of the true function as part of the optimisation, retraining the surrogates periodically throughout the process [31–33].

[1] A non-active constraint is one which is eminently feasible, and is not limiting the objective function. An active constraint is any factor which is causing the limitation on the objective function, those which are the closest to being infeasible.

Surrogate models are sometimes referred to as 'meta models' or 'response surface models' within the literature, usually dependent on how they are generated. Generally all feature a representative function, $\tilde{f}(x)$, which emulates the behaviour of the true function, $f(x)$, whilst being computationally cheaper to analyse.

Surrogate assisted optimisation has made the design of expensive problems simpler, the key being that they are more accurate in the region of interest rather than having a high level of accuracy across the entire design space [34, 35, 35, 36]. Efficient large-scale structural optimisation using surrogate analogues to approximate a multilevel structural model have been successfully tested but not bench-marked against existing architectures [37].

The most widely cited form of surrogate assisted optimisation within the literature is *Efficient Global Optimisation* (EGO). This method fits response surfaces to data collected through sampling with both prediction and error estimations to guide an infill sampling criterion towards promising areas. This method must be tuned to best strike a balance between improving the global surface properties and improving the local approximations about the minima. Note this method in its original form can only be applied to a relatively small number of input variables because of the kriging limitations in higher dimensions. [38].

Multi-Start and **Global Search:** For problems with multiple local minima, the simplest approach is to simply use multiple start points. These algorithms explore multiple regions of the design space to find globally optimal solutions [25, 39].

Multi-start procedures simply produce a set of N_{start} positions, either uniformly distributed or space filling start points to be optimised in-turn. This approach guarantees convergence to a globally optimal solution as N_{start} converges to ∞. However, this procedure can be inefficient as the same local solutions may be located many times for additional computation.

Global search procedures reduce the sampling space of the solution based on the boundary of the previous locally optimal solution. A local optimiser finds the ith local minima v_i^*. The boundary is defined by a *basin of attraction*, $B(v_i^*)$, determining a set of starting points S, from which the optimiser converges to v_i^*, equation (1.4). The global phase searches the remaining solution space using a scatter-search method, a deterministic population-based solver, to find potential starting positions from a smaller design space [40].

$$v_i^* : B(v_i^*) = \left\{ \underset{S}{\operatorname{argmin}} \left(f(v_0) = v_i^* | v_0 \in S \right) \right\} \tag{1.4}$$

Global search procedures offer significant improvements in efficiently searching the design space and an important tool for solving real-world problems [41]. Difficulty increases as the number of minima within the design space increases as each potential minimum present within the solution space results in another optimisation, until the complete design space has been assessed. If the design space is highly multi-modal or features plateaus these procedures can result in a significant amount of computation.

Stochastic global phases have been used to improve the responsiveness for highly dimensional problems, but this only reduces the likelihood of the global-phase search finding repeat optima [42]. If basins of attraction for each optimum are significantly smaller than the design space, assessing every potential optimum can still be prohibitively inefficient when compared to more direct stochastic optimisers.

1.2 Multilevel Optimisation

MLO is the process of partitioning a single system into a set of decoupled subproblems, within a single system coordination problem. Any multi-model problem where there is precedence or uni-directional transfer of information between models is an MLO problem. The ubiquity of these types of problems has meant there are a wide array of applications across many areas of research. Figure 1.1 combines the qualitative analysis of a range of survey papers, each compiled an exhaustive discussion of methods and theories into a research map of topics, and subtopics. The major subsets identified within the MLO literature are *hierarchical design*, *networked systems*, *cooperative behaviour*, and purely *theoretical research* [15, 43–47].

Networked Systems: Domain decomposition initially focused on mathematical programming for large-scale linear programs, but later with the developments in computing power, this widened to include parallel processing and network partitioning. Networked systems[2] has been one of the earliest and most impactful areas of research within the literature since the 1950s [48]. Modern applications have encompassed phone and network utilisation problems, transit, logistics and freight routing optimisation problems.

Network design problems are noted for being non-linear, non-differentiable problems. Design problems usually result in numerical models with either mixed-discrete or fully combinatorial design/ solution domains. The nature of networked problems has led to a range of research into stochastic solvers and other derivative-free methods including the use of GAs with hierarchical populations, or additional clustering and sharing functions [49–51].

Theoretical Research: Due to the potential incentives for more accurate solutions to the non-linear decomposed or network multi-model problem there is a wide-base of mathematical and theoretical solutions. However, classification and verifying the global optimality of subsequent solutions can be difficult, and often problems are classed as NP-hard[3]. The diversity and complexity of problem hierarchies involved means there is no consensus on the 'best' overall method [8, 53].

[2] Networked systems refers to any system which features two or more computers or nodes which are linked together for the purpose of completing a joint goal.

[3] NP-hard refers to Non-deterministic Polynomial-time hardness, a classification of algorithms defined by their difficulty; any computational decision problem within this class are at least as hard as the hardest decision problems for which a verifiable solution exists [52].

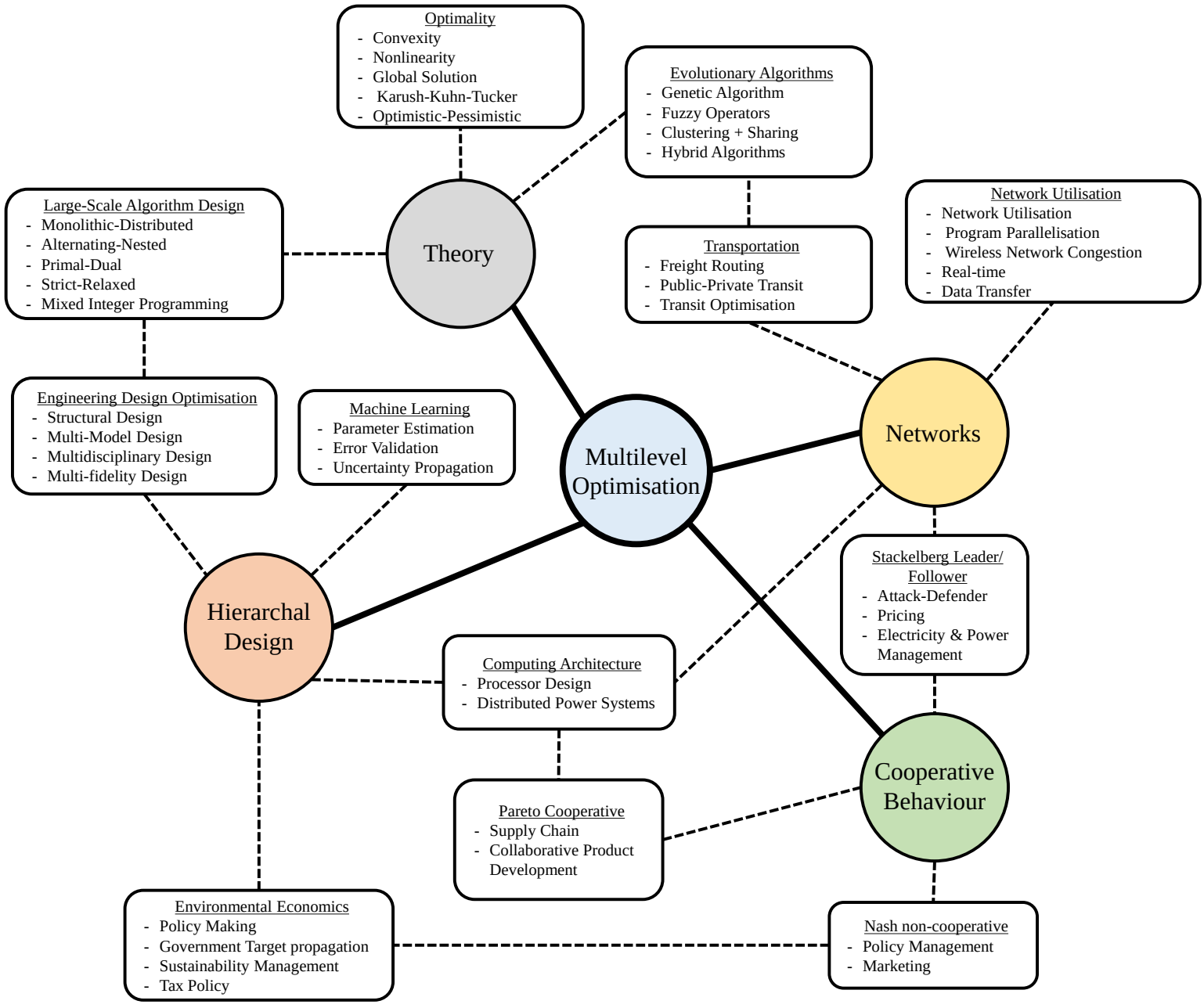


FIGURE 1.1. Multilevel optimisation research map; fields of study and connections between various application and theories since the 1950s.

Development of MLO algorithms has expanded with a diverse number of applications, growth in computing power, smarter algorithms and the need for designers to pose more substantial optimisation problems. This expansion has been pronounced in the fields of behavioural models and hierarchical designs, leading to a significant surge in the breadth of literature available.

Cooperative Behaviour Models: So far, the discussion has been focused mainly on MLO as a tool for design, however, MLO has been most beneficial in the prediction of how people, companies and countries respond to each other. Descriptive models of multi-agent[4] behaviour are based on the Stackelberg competition and problems are defined within multi-model hierarchies. Upper-level decision-makers are cast as a *leader*, manager, attacker, or governing system; optimising decision vectors to suit broader goals. The lower level responding entities exist within or in response to actions at the previous level and with individual autonomy; as a *follower*, worker, defender or local entity [54].

These methods have a considerable interest outside of engineering and programming in the fields of economics, management, public sector decision making and defence. Often cooperative behaviour models are referred to as a class of "Bi-level Programming Problems" (BLPP).

Hierarchical Systems Design: An upper level system model is coupled with subsystem and component level models, providing additional detail to inform overall design. MLO is used to propagate system-level targets to local subproblems aimed at enforcing localised constraints. Figure 1.2 shows a typical hierarchical layout, where the product has been decomposed into a modular set of system, subsystem and component level problems [55].

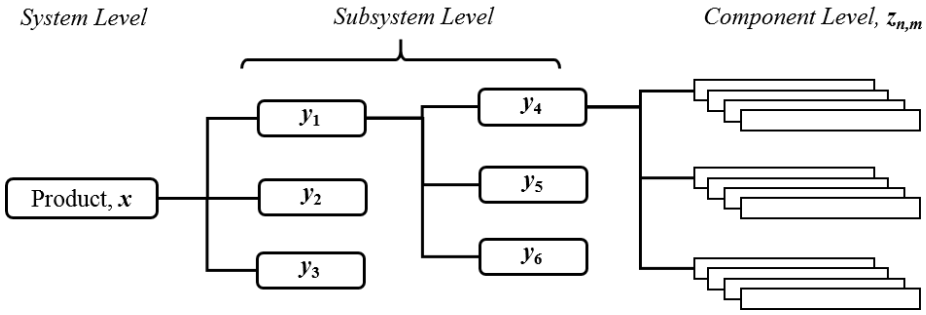


FIGURE 1.2. Hierarchical System Layout; General design Decomposition into a hierarchy of $\boldsymbol{n}$, subsystems with $\boldsymbol{m}$, components.

Engineering applications have been particularly diverse including: network/simulation partitioning, bio-engineering, supply chain/process engineering and hierarchical systems design, such as in circuitry and structural design. These methods have branched out to include additional fields such as image processing, machine learning and multi-fidelity optimisation [56, 57].

[4] Agent refers to an independently operating person, business or entity.

Design problems decomposed into MLO problems are functionally the same as single objective MDO problem. Both require a statement of a multi-model problem to be solved with the only significant difference being that subproblems must be arranged into a hierarchy [8].

Although the MLO problem is found in many instances of design, not all complex systems can be solved through decomposition. The general characteristics of an MLO problem require:

(a) *Acyclic*[5] *Systems Design*; meaning that sub-components must form a hierarchical or tree-like structure, such as the system shown in figure 1.2.

(b) *Enforcement constraints*; for each constraint disseminated into a subproblem there must exist some variables that associate it to the system-level problem.

(c) *Equivalence of copies*; variables which exist or are associated to several other variables in multiple subproblems must be consistent. To maintain the primal[6] form of the problem there must be some binary constraint such that all "copies" of variables assume the same value across all levels.

The next section will focus on aspects of decomposition theory, highlighting salient methods and classifications used in both hierarchical design and cooperative behaviour models.

1.2.1 Decomposition Theory

So far, the literature has identified two areas of research, MLO problems and BLPP. MLO problems are multi-model problems, with an upper/leader and follower/lower level problem, with each acting independently from one another. BLPP decision-making models of policy where leaders and followers are entities of principle agency, such as companies, government agencies or people. Both BLPP and MLO problems are non-convex and non-differential, where given subproblems result in the presence of multiple local levels of optimal solutions [59].

The nature of the relationship between the leader and follower can vary between systems. The local objectives and constraints of each model can be some combination of cooperative or competitive inter-model relationship, depending on the predicted system behaviour changes can result in very different algorithmic requirements. As assumptions in BLPP, we assume that cooperative or competitive indicates either an *optimistic* or *pessimistic* view of the multi-model system, respectively. In both cases, the final solution is lower-level feasible and therefore, a valid optimal solution. The classification of optimistic or pessimistic relates to the lower level decisions impact on the upper-level system. In other terms:

[5] Acyclic meaning not displaying or forming part of a cycle.
[6] Primal referring to the original form of the problem, before being partitioned [58].

Optimistic: Assumes the subproblem selects the solution which also leads to the best outcome for the upper level problem. Models cooperate without limiting constraints or overly conflicting objectives.

Pessimistic: Assumes the subproblems solution has selected the worst-case result for the upper level problem. Models compete, local model feasibility and optimality runs counter to the overall system goals.

Luckily the engineering MLO design problem is more tractable than some cooperative multi-agent BLPP. The partitioned engineering design problem does not feature a set of independent agents, instead it has a number of local parts and components which must be coordinated towards a single goal. In this sense measures of optimism' and 'pessimism' are not defined positions based on individual ideology. Instead they are measured by the competitiveness between inter-model relationships.

An optimistic position occurs in engineering design when the level of competitiveness between subsystem models is reasonably non-intrusive or limiting to the overall system behaviour. A pessimistic position can occur when the competitive relationship between subsystem models acts as a significant hindrance to the system level goals or detrimental to adjacent system level responses.

For example when the problem is to minimise structural mass and maximise performance, reduced mass anywhere in the system supports that goal i.e. the entire system has a cooperative 'best' solution resulting in the best system-level response. Note this does not necessarily indicate local subsystem optimal designs are automatically best for the overall system, competitive or intrusive behaviours can occur when there are significant differences in modelling domain or differing disciplines. For example, a shape-based aero-structural optimisation using an upper-level aerodynamic model, and lower-level structural model; in this case the optimal aerodynamic solution differs from the optimal structural solution, resulting in a competitive relationship toward different optimality conditions based on designer requirements. Defining whether a design relationship is optimistic or pessimistic is dependent on whether optimal subsystem compliance to local constraints reinforces or is detrimental to the upper system-level problem.

Within aerospace and structural design, the earliest work comes from Sobieszczanski-Sobieski, 1985 [7]. This work focused on a two-level gradient-based method, where individual subproblem optimisation precipitated optimisation of the assembled structure. Top-level system targets were propagated through a hierarchy to ensure a feasible design; the two-level approach was tested and verified against results obtained from a conventional single-level optimisation. The format of developing a new optimisation architecture and testing against a conventional baseline has since become the standard with a significant body of work solving more elaborate versions of the same general problem.

1.2.2 The General Problem

The breadth of research available for the decomposed design problem has led to the development of a wide range of techniques and general multi-model architectures within the literature [15, 43, 46]. This section will first outline the general MLO problem and then classify the range of methods designed to solve it.

The simple optimisation problem in equation (1.1) can be decomposed into an MLO problem, given in equation (1.5). The model now has a system-level problem, $F_0(\boldsymbol{x},\boldsymbol{x}_0)$, and N local subproblems, $f_i(\boldsymbol{x}_0,\boldsymbol{y}_i)$.

The sub-vector, $\boldsymbol{x}_0$, represents *linking variables* those which exist in multiple subproblems. Without linking variables, F_0 and $f_{1,\cdots,N}$ could be optimised independently to find the global minimum $F(\boldsymbol{v}^*)$. In this formulation the lower-level problem is a constraint, such that only those members considered lower level optimal may satisfy the upper-level problem.

$$\begin{aligned}
\underset{x,x_0}{\text{minimise}} \quad & F(\boldsymbol{v}) = F_0(\boldsymbol{x},\boldsymbol{x}_0) \quad \text{with respect to} \quad [\boldsymbol{v}] = [\boldsymbol{x},\boldsymbol{x}_0,\boldsymbol{y}_{1,\cdots,N}] \\
\text{subject to} \quad & c_0(\boldsymbol{x},\boldsymbol{x}_0) \leq 0 \\
& arg \min_{x_0,y_i} \{f_i(\boldsymbol{x}_0,\boldsymbol{y}_i) : c_i(\boldsymbol{x}_0,\boldsymbol{y}_i) \leq 0 \in i = 1,\cdots,N\} \\
& \boldsymbol{v}^{\mathrm{L}} < \boldsymbol{v} < \boldsymbol{v}^{\mathrm{U}}
\end{aligned} \tag{1.5}$$

Here $\boldsymbol{v}$ represents the combined design variable vector $[\boldsymbol{x},\boldsymbol{x}_0,\boldsymbol{y}_{\boldsymbol{1},\cdots,\boldsymbol{N}}]$ corresponding to the system-level, linking and subsystem-level design variables respectively. c_0 represents the system-level constraints. f_i and c_i represent subsystem-level objectives and constraints for N number of subsystems. v^{U} and v^{L} represent the upper and lower limits respectively on decision variables in the positive real domain, $\boldsymbol{v} \in \mathbb{R}$. All following problems are assumed to exist within the same bounds and domain as defined for equation (1.5).

When linking variables are the only information shared between subproblems, i.e. there are no associated multi-system dependent objectives or constraints, the decomposed problem is known as a *quasi-separable function*. The shape, sizing structural design problem presented will be based on quasi-separable assumptions. The overall shared objective will be for each localised substructure to reduce the overall mass of the assembled structure.

1.2.3 Optimisation Architectures

Figure (1.3) is a general overview of the range of MLO solvers available with the literature. MLO solvers can be grouped into the same major subsets as those used in MDO: *Monolithic* and *Distributed* architectures, with additional distinctions based on the attributes of each solver [15].

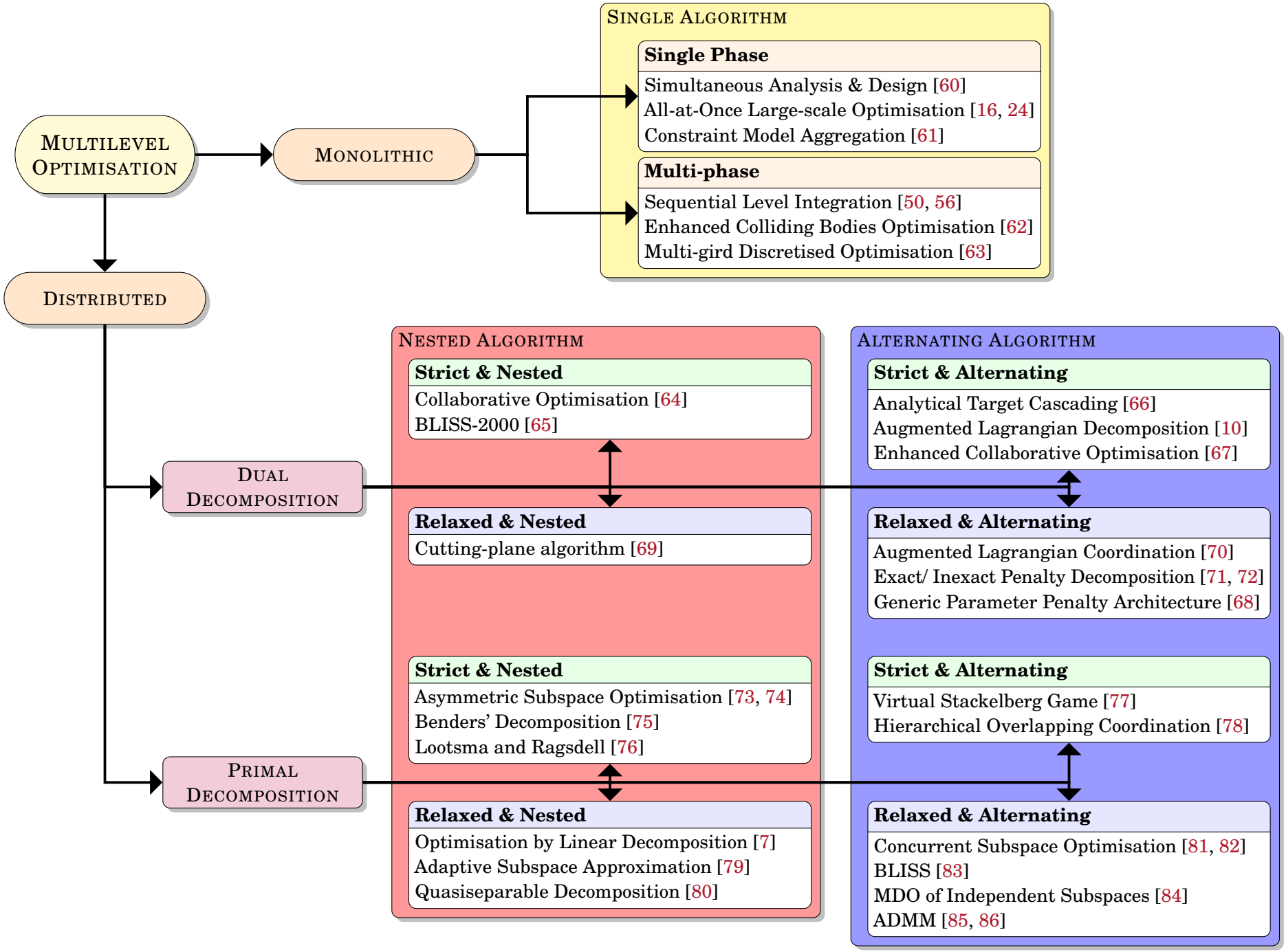


Figure 1.3. Multilevel Optimisation Architectures; with classifications based on the type of algorithms presented.

1.3 Monolithic Approaches

The monolithic optimisation problem combines all models, variables and constraints into a single design loop — one of the more straightforward approaches to directly solving equation (1.5). Equation (1.6) is the result of concatenating the MLO problem statement into a single design problem, this can then be solved through either a single-phase or multi-phase process.

$$\begin{aligned} \underset{v}{\text{minimise}} \quad & F(\boldsymbol{v}) = F_0(\boldsymbol{x},\boldsymbol{x}_0) + \sum_{i=1}^{N} f_i(\boldsymbol{x}_0,\boldsymbol{y}_i) \quad \text{with respect to} \quad [\boldsymbol{v}] = [\boldsymbol{x},\boldsymbol{x}_0,\boldsymbol{y}_{1,\cdots,N}] \\ \text{subject to} \quad & c_0(\boldsymbol{x},\boldsymbol{x}_0) \leq 0 \\ & c_i(\boldsymbol{x}_0,\boldsymbol{y}_i) \leq 0 \quad \text{for } i = 1,...N \\ & v^{\mathrm{L}} < \boldsymbol{v} < v^{\mathrm{U}} \end{aligned} \tag{1.6}$$

1.3.1 Single-Phase Solvers

The simplest optimisation architecture presented, but also the most widely used and verifiable. The single combined optimisation problem is solved in a single-phase 'All-at-Once' (AAO). The single-phase approach does limit the scope of problems as it requires all variables, including local individual subsystem variables to be considered, assembling into equation (1.6). Initial work used the *simultaneous analysis & design* of all multi-model objectives and constraints. However, resulting problems were too large to be efficiently solved using stochastic optimisers without becoming prohibitively computationally expensive; while gradient-based solvers can result in poorly scaled problems within multi-model non-convex solution spaces. Even once design variables are normalised, the localised and nuanced nature of component and part-level variables/ responses can lead to poorly scaled optimisation problems [60].

Gradient-based solvers are the only way to guarantee an effective solution to any large-scale problem. Convergence properties for the most challenging problems have been improving with the use of more modern algorithms. With sufficiently advanced adjoint methods and quadratic programming larger higher dimensional MDO and non-convex problems have been solved within the literature [16, 24].

By optimising all variables and constraints together, monolithic methods present the most accurate definition of the complete un-partitioned design problem. Monolithic methods do not need to account for the interdependence of quasi-separable subproblems because all variables exist within the same problem and the solution uses the most accurate representation of the combined behaviour. The original (primal) form of the problem is unchanged without the need to manage linking variable consistency artificially [15].

The interdependence of structural sub-components allows for some localised off-optimal designs in-order to improve the overall system behaviour. For example in structural design, allowing the mass of components in certain areas can manipulate the overall load distribution

across the entire structure, alleviating more pressing constraints elsewhere and leading to overall weight savings.

Furthermore, there has been a wide range of research into constraint aggregation and efficient sensitivity analyses making the processing and usage of large-scale monolithic solvers within MDO, particularly mature [16, 61]. In benchmark tests, modern monolithic architectures using gradient-based adjoint solvers are faster and more efficient than stochastic solvers for sufficiently large problems. Stochastic methods scale the number of function evaluations quadratically with the number of design variables, gradient-based solvers scale linearly, resulting in larger differences in efficiency based on the problem size [22].

On the other hand, accurate gradient evaluation can cause a results bottleneck limiting the size of problems. A gradient can be found either through the differentiation of an analytical problem or finite differencing[7]. If finite differencing is used, despite being the simplest to implement, can lead to a marked increase in computational expense, especially if the function evaluation involves external black box solvers, CFD or FE models. This is because for every design step the solver would need to run all models with respect to every variable

1.3.2 Multi-phase Solvers

One can improve the efficiency of monolithic solvers by gradually increasing the complexity of the design problem as the solution becomes more apparent. These have been defined here as sequential methods (SEQ). By gradually increasing the scope of the optimisation problem into a series of phases or associated levels it is possible to mitigate the cost of running every subsystem model from the very beginning. The partial solution from each phase is then used as the starting position for the one next, and so on until the design is complete. Phases of design are based on a range of factors, including but not limited to computational expense, elaborate meta-heuristic[8] processes [62], or manually defined functions of refinement [63].

Levels of design are often defined using associated computational expense, this improves efficiency by ensuring models with the highest associated costs are used sparingly during the optimisation. SEQ methods gradually increase the difficulty of the model and they are relatively easy to implement. They have been used extensively within multi-fidelity optimisation where more elaborate or higher resolution functions are only referred to in the latter phases of the solution. This procedure is the most efficient when using discrete level mixing when tested against systems of gradual mixing or clustered heuristic algorithms [50, 56].

[7] A process used to approximate derivatives of a function; it is most often applied by systematically calculating the difference quotient for each dimension in space by some fixed distance from the current design point.

[8] Any approach to problem solving not guaranteed to be optimal but sufficient for reaching an immediate goal [87].

1.4 Distributed Approaches

Through partition and decomposition, distributed approaches convert large problems into a set of smaller isolated and numerically determinate subproblems, with each can then be solved individually. subproblems are then subject to additional global coordination constraints, dictating how sub-components fit together in an iterative algorithm.

Distributed solvers are designed for problems where it is currently considered impractical, computationally expensive or impossible to contain all design features to a single function evaluation [88, 89]. The development of distributed methods closely mirrors the existing industrial engineering environment. The ideal solver would combine disparate and specialist teams, often with unique black-box solvers, into a robust network of distributed optimisation process [90].

For numerical test problems, the only comparative metric where distributed methods improve on monolithic solvers was in *portability*[9] — the ability to fit into existing systems, measured by the division of labour and design autonomy. Monolithic methods are superior in terms of simplicity, transparency, efficiency and accuracy. However, comparative studies often focus on numerical problems with feasible and convex solution spaces, so it remains unclear whether the improved accuracy of monolithic methods is consistent for more realistic models under a range of scenarios [12].

Algorithms can be classified based on the consistency, feasibility and formulation methods used during each phase. The subdivision of distributed solvers is as follows [13]:

- **Consistency:** Based on the handling of linking variables [58].
  - *Primal decomposition* is when linking variables remain consistent between all models throughout the optimisation process. Methods maintain the primal form of the problem, with parity between linking variables consistently enforced amongst subproblems for each design iteration. Variable parity can be achieved either through the use of equality constraints tied to specific variables, or narrowing local partitions to ignore linking variables. This is sometimes referred to as closed inter-model consistency.
  - *Dual decomposition* relaxes consistency constraints, allowing local copies of linking variables to be independently modified within subproblems. Consistency is often enforced using a weighted quadratic penalty method for inconsistent designs, only restricting local design space exploration in the later stages of design. They are sometimes referred to as open inter-model consistency.

[9] The ability for an object or method to be transferred from one system to another.

- **Formulation**: Based on the order of system and subproblems optimisation phases [13].
  - *Nested algorithms* are BLPPs. Decomposed subproblems exist within some outer system-level or coordination problem. With each design-step of the outer problem, a set of inner subproblems are solved.
    The resulting outer problem may pose a problem for gradient-based solutions as the design space has no guarantee of convexity, and the KKT conditions are not necessary for optimality.
  - *Alternating algorithms* iterate between solving a system problem and subproblems in sequence. A coordination penalty method or design space narrowing ensures convergence to a consistent result toward the end of the optimisation process. Local convergence proofs are available for these methods, but it does require strict assumptions on local design separability.
- **Feasibility**: Based on the enforcement of design constraints with each iteration [14].
  - *Strict methods* require that all design constraints are satisfied, and optimality conditions are met for each iteration. These are most often applied to strictly convex problems where a possible solution must exist within the design domain. Sometimes strict is referred to as a closed design method.
  - *Relaxed methods* allow for open design constraints and focus on consistency across subproblems. Usually some ramp or prohibitive penalty functions ensures final design feasibility. Artificially lowering design constraints is nontrivial and can radically change the position of design optima, meaning algorithms usually require significant tuning. Most relaxed methods simply allow for non-feasible solutions to be accounted for during the next phase of the optimisation process.

By breaking the analyses into stages distributed methods reduce the size of an optimisation problem to the scope of the largest defined subproblem enabling efficient exploration of the design space. As long as one can continue to partition models accurately, any number of constraints can be added to the optimisation problem without making the problem numerical indeterminate [91]. Solvers using gradient-based methods can outperform monolithic solvers under a range of different conditions; however, this is reliant on whether there is a large enough computational discrepancy between individual subproblems. When one subproblem is significantly larger than others, the most efficient algorithms can use these more computationally expensive techniques sparingly, improving efficiency. The more significant the discrepancy, the more capability for the designer to make performance improvements through a multi-phase or distributed algorithm [73].

1.4.1 Dual decomposition

Dual decomposition allows local 'copies' of linking variables to be used in subproblems, allowing them to be solved in isolation. This strategy allows for diverse teams or the use of black-box solvers, using a potentially generic and robust external coordinated optimisation routine. The coordination problem is then to match up the average values of linking variables between subproblems. Implicit coordination between subproblems can be achieved either using additional *consistency constraints* to replicate the coupling between subproblems, or some domain restricting effect achieved through *penalty relaxation*, penalising solutions with system: subsystem discrepancies.

The resulting system and subsystem level problem statements using a weighted consistency constraint are given in equations (1.7) and (1.8):

$$\begin{aligned} \underset{x,x_0}{\text{minimise}} \quad & F_0(\boldsymbol{x},\tilde{\boldsymbol{x_0}}) \\ \text{subject to} \quad & c_0(\boldsymbol{x},\boldsymbol{x}_0) \leq 0 \\ & w_0.q_0(\boldsymbol{x_0},\tilde{\boldsymbol{x_0}}) = 0 \end{aligned} \tag{1.7}$$

$$\begin{aligned} \underset{\tilde{x_0},y_i}{\text{minimise}} \quad & f_i(\boldsymbol{y}_i,\tilde{\boldsymbol{x}}_0) \quad \text{for } i = 1,...N \\ \text{subject to} \quad & c_i(\boldsymbol{y}_i,\tilde{\boldsymbol{x}}_0) \leq 0 \\ & w_i.q_i(\boldsymbol{x}_0,\tilde{\boldsymbol{x}}_0) = 0 \end{aligned} \tag{1.8}$$

Here $\tilde{\boldsymbol{x}}_0$ represents the local variable copies which must exist with the same bounds and domains as the originals. q defines new equality consistency constraints. Parameters w_0 and w_i, represent some consistency weighting system or coupling function.

Nested formulation: Nested methods using dual decomposition are the least common combinations of methods present in the literature. Nested algorithms complete a full set of optimisation subproblems with each step of the outer loop function, the result being that there are a significant number of local function evaluations. Any increase in difficulty or complication to local subproblems exponentially increases the computational cost of the entire process, especially when including additional coupling mechanisms.

Some of the earliest examples of nested algorithms using dual decomposition was the *Cutting-plane algorithm* from Balling & Sobieszczanski-Sobieski, which focused on numerical solutions [69]. Strict convexity and optimality conditions are necessary for the most successful versions of these algorithms. Any particularly complex or unsolved subproblems can dramatically increase the number of outer-loop steps.

Within design optimisation the optimal solution often lies on some constraint boundary; a design is optimal when most, if not all, constraints are active. This is indicative of a design that does not feature any wastage or additional room for improvement.

If the inner-loop steps cannot converge successfully the closer they get to a possible optimum the algorithm may become erratic, leading to non-smooth design steps in the outer loop problem. The resulting outer-loop solution space will feature discontinuities about the global optima. Later algorithms would include tests on limited simulation and design-based models.

In *Collaborative Optimisation* [64, 92] (CO) the system-level problem is optimised for some minimum objective, while subproblems minimise system-sub problem discrepancies. CO has been successfully coupled with multi-fidelity modelling and a combination of global approximations and trust region strategy at the system level in-order to improve the effectiveness of this method. The result was a comparable result to monolithic methods and a significantly improved level of convergence and computational efficiency when compared to the original method [93].

CO was later improved, converting it into an alternating formulation called *Enhanced Collaborative Optimisation* (ECO), which reverses the roles of the system-level problem and the subproblems, so that the system-level problem minimises inconsistency, while the subproblems minimise system objectives. ECO was effective in improving the efficiency of the original algorithm reducing the number of subproblem model evaluations required for a converged solution [67, 94].

Alternating formulation: The dual-alternating algorithm is one of the most common algorithms in the literature. The resulting problem statement is applicable within a range of research areas, especially those where there is an optimistic problem relationship and minimal linking variable deviation. Alternating methods can also be readily applied to problems with L levels of problems, as unlike nested problems the number of function evaluations does not increase exponentially with each new level.

If there are no system-wide constraints or objectives, i.e. neither F_0, c_0 exist, *Exact* / *Inexact Penalty Decomposition* can be used [71, 72]. In these algorithms each optimisation subproblem is optimised with respect to a quadratic penalty function enforcing linking variable converge to the optimal design.

One of the most popular architectures in structural design is *Analytical Target Cascading* (ATC) [66]. ATC uses weighted multipliers applied to consistency penalty functions, with weights increasing until a desired consistency is achieved, as shown in equations (1.7) and (1.8). System-level targets can then be readily applied to bi-level and multilevel decomposition schemes, without the need for subproblems coupling, allowing self-contained analyses of subproblems.

The resulting decomposed coupled subproblems are solved through an iterative process. The aim is to reduce consistency penalties and thus solve the primal form of the problem while finding an optimum for the coupled problem. The maturity of this algorithm has meant there have been several updates to the original methodology. The most efficient method presented combines several other methods, using an alternating direction method of multipliers to update Lagrange multipliers applied to penalty parameters [10, 70, 95]. The resulting generic augmented ATC algorithm allows for both strict and relaxed methods [68].

A range of convex product design problems, including aircraft and automotive examples have used ATC in the past [96–101]. ATC is often stated as a practical alternative for initial concept design in the case of unconventional aircraft configurations when historical data or previous experience is not available [102, 103]. In some studies ATC is seen as so robust and accurate, its used as the benchmarking algorithm for a range of novel distributed optimisation architectures [68, 104].

When tested against alternative distributed architectures, both ECO and ATC achieve rapid and accurate solutions when compared to a single-phase monolithic optimisation. Most comparative studies between alternating and nested methods find comparable results, with variations based on algorithmic distinctions, impacting the types of problems solved [105, 106]. Some problems are more readily adaptable to either the ECO, ATC or AAO methods depending on the initial design assumptions. These include quasi-separable assumptions, formulation, convexity, implementation of target propagation and the likelihood of either optimistic or pessimistic conditions (level of system: subsystem competitiveness).

The weighting update strategy and limits can significantly change the efficiency of ATC based methods. The most efficient methods presented uses the sub-gradient to solve for the dual Lagrangian problem [10, 107] with a range of convergence proofs for this class of problem [78, 108]. For the general two-level case, any implicit non-linear equation with multiple concurrent subproblem optimisations converges to the same KKT condition as the monolithic problem [109]. Under quasi-separable assumptions, the general dual Langrangian form of ATC simplifies to the augmented ATC algorithm [110].

1.4.2 Primal decomposition

Primal decomposition methods distribute the analysis into several subproblems but maintain the unpartitioned consistency of the design problem. Linking variables remain consistent amongst subproblems, with some outer process supplementing the coupled optimisation problem. Subproblem coupling within the literature can be broadly arranged into three main techniques, using: *Interaction Approximation Methods*, *Coupled Optimality Methods* or a *Virtual Stackelberg Game*.

Interaction Approximation Methods: Linking variables remain fixed as part of the subproblem optimisation, deriving partial optimal values with respect to the sub-vector, $\boldsymbol{x}_0$. Variables within the system-level coordination problem then use these partial optimal values. By recasting subproblems with a parametric function, an interaction approximation model, one can couple the coordinated problem through a series of parametric functions.

One of the earliest primal MLO methods using interaction approximation models was a relaxed consistency alternating formulation *Optimisation by linear decomposition*. These methods were later developed into the *Bi-level Integrated System Synthesis* (BLISS) architecture [83]. BLISS nested artificial interaction models for subproblems, allowing one to solve the resulting

coupled upper-level problems using a suitable single-phase optimisation technique. The primal decomposition system-level problem statement is given by equation (1.9).

$$\begin{aligned} \underset{x,x_0}{\text{minimise}} \quad & F_0(\boldsymbol{x},\boldsymbol{x}_0) + \sum_{i=1}^{N} \tilde{f}_i(\boldsymbol{x}_0, \hat{\boldsymbol{y}}_{\boldsymbol{i}}(\boldsymbol{x_0})) \\ \text{subject to} \quad & c_0(\boldsymbol{x},\boldsymbol{x}_0) \leq 0 \\ & \tilde{c}_i(\boldsymbol{x_0}, \hat{\boldsymbol{y}}_{\boldsymbol{i}}(\boldsymbol{x_0})) \leq 0 \quad \in i = 1,...N \end{aligned} \tag{1.9}$$

$\tilde{f}_i$ and $\tilde{c}_i$ represent artificial functions depicting partial subsystem-level solution, $\hat{y}_i$, for problems $i = 1,...,N$, with respect to $\boldsymbol{x}_0$. The intermediary step between the system and subsystem level is the generation of some approximate model representing the function given in equation (1.10).

$$\left.\begin{aligned} \tilde{f}_i(\boldsymbol{x}_0, \hat{\boldsymbol{y}}_i(\boldsymbol{x}_0)) &\models f_i^*(\boldsymbol{x}_0, \boldsymbol{y}_i^*) \\ \tilde{c}_i(\boldsymbol{x}_0, \hat{\boldsymbol{y}}_i(\boldsymbol{x}_0)) &\models c_i^*(\boldsymbol{x}_0, \boldsymbol{y}_i^*) \end{aligned}\right\} \quad \text{for } i = 1,...N \tag{1.10}$$

f_i^*, and $\boldsymbol{y}_i^*$ represent the optimal values for any given input $\boldsymbol{x}_0$. The subsystem-level problem takes the simplest form, equation (1.11), and is used as an input to generate the approximate function model above.

$$\begin{aligned} \underset{y_i}{\text{minimise}} \quad & f_i(\boldsymbol{x}_0, \boldsymbol{y}_i) \quad \text{for } i = 1,...N \\ \text{subject to} \quad & c_i(\boldsymbol{x}_0, \boldsymbol{y}_i) \leq 0 \end{aligned} \tag{1.11}$$

Primal methods work best when there are relatively few linking variables, and the solution has fast solving subproblems to iterate through. Within structural design, detailed substructure models tend to be used for subproblems, hence generating surrogate models can be time-consuming and computationally expensive. When the subsystem-level problems are sufficiently large it is possible to use level integration schemes, partially updating post-optimality derivatives with respect to linking variables, $\boldsymbol{x}_0$, throughout the solution. Periodic updates to interaction models reduce the number of evaluations for computational expensive models by focusing model precision on specific areas of the design space, about which a global optimum can be found [56, 111].

The original format of BLISS used partial-optimality sensitivity analyses to allocate subgradients for approximate variation in subsystem behaviour with system-level changes. Effectively sub-gradients used at the system-level for local constraints relative to linking variables, $\frac{dc_i}{dx_0}$ [7]. The *Asymmetric Sub-Optimisation* approach uses a similar partial optimality approach and under a range of conditions outperformed monolithic solvers in engineering design problems. However, both these methods require a coupled most-optimality sensitivity method and the second derivative of those design sensitivities. This approach can lead to significant post-processing and bottlenecks in the computation for higher-dimensional problems [73].

BLISS-2000 was an improved form of BLISS, switching to a dual formulation which maintained design feasibility by replacing sub-gradients with preallocated surrogate models [65]. BLISS is not competitive with other MDO architectures in terms of computational costs because of the additional sub-gradient sensitivity analyses. BLISS-2000, on the other hand, has been found to outperform other MDO algorithms and can be competitive when compared with monolithic AAO methods. By changing to a dual decomposed form of the problem, and coupling additional consistency constraints with surrogate-based interaction approximation models, BLISS-2000 was able to outperform monolithic solvers in numerical benchmarking studies [15].

Coupled Optimality Methods: A potentially generic solution to the problem of coupling problems in primal decomposition is to use the 'coupled post-optimality sensitivity' (CPOS) method. The aim is to create a set of aggregate functions, overlapping the optima from the system and subsystem levels. First one can replace subsystem models in the coupled problem by assuming the optima within local subproblems satisfy the KKT conditions. Instead of using an approximate model of local behaviour, the approximate model directly replicates the variation in local optima with respect to linking variables. The reformulated and aggregated subproblems then take the same format of equation (1.10). With the local problems all synchronised to the aggregate problem, each can be solved using a distributed optimisation. CPOS is the method of coupling used in *asymmetric subspace optimisation* algorithm [73, 74, 112, 113].

However the problem with CPOS based methods is they have strict conditions for application. An optimisation problem can be solved using CPOS if:

(a) For the optimal feasible subset of solutions, $\boldsymbol{y}_i \in f_i$ and $\boldsymbol{y}_i \in c_i$ can both be differentiated twice, with a strictly continuous, convex and compact solution space.

(b) The resulting approximate optimal function $\tilde{f}_i(\boldsymbol{x}_0, \hat{\boldsymbol{y}}_{\boldsymbol{i}}(\boldsymbol{x_0}))$ is also a real and continuous function, where the resulting global solution is defined as seen in equation (1.11) [114].

The requirement to have a twice-differentiable, real and continuous design space is often too restrictive a requirement to make CPOS and similar methods generally applicable. Assuming a radius of convex solution space about the global optima could alleviate this requirement, but requires that the algorithms starting point to be sufficiently close to said optima. This assumption allows one to use a convex-specific gradient-based solver locally when the surrounding global solution space is non-convex. Many multi-start and stochastic-gradient hybrid methods use this assumption [40]. However, the twice-differentiable function requirement can be too computationally prohibitive to use with most analytical models or black-box solvers.

Virtual Stackelberg Game: The simplest subproblem coupling method, in the broadest terms, iterates between the subproblem and system-level until a stable equilibrium is reached or the solution is unchanging. This method has been used in a range of engineering research because it is easy to set-up and test, with minimal tuning parameters or modelling assumptions. Within the literature the most formative elements, advances and unique variations of these methods have been developed for environmental economics, where it is known as the *Stackelberg leader/follower problem* [54].

Engineering research using MLO has focused on iterating between fully separable global: local methods. Multi-model structural designs in most cases iterate to a stable equilibrium. However, there is no mathematical assurance of convergence to a coherent global solution through iteration alone [37, 102, 103, 115–120].

The *adaptive sub-space approach* is a nested primal implementation of a virtual Stackelberg method[10]. An iterative method which updates subproblem partitioning based on variable dependence. Reduced dimensional surrogates are then generated within some trust-region about the current design point. With each design step of the system-level problem the nested subsystem-level problem is solved within the defined trust-region.This method found that reasonable fixed partitions relative to external parameters can outperform automatic variable dependency-based partitions within an aero-structural design environment [79].

Alternating Direction Method of Multipliers (ADMM) is a relaxed alternating formulation first introduced to solve the general multi-model leader/follower problem using a coupled Lagrangian approach. Effectively an aggregate of the multi-model problem, $\bar{L_p}$, is created using slack variables for lower level inequality constraints and Lagrangian multipliers for problem scaling. The leader, follower and slack variables are optimised in-turn, with an outer loop updating Lagrangian multipliers. This method reduces the use of state variables and assumptions in stating the optimisation problem [85, 86]. Excessive state variables within architecture can lead to a substantial amount of problem tuning and poor performance in a novel or unique case.

The resulting aggregate problem, $\bar{L_p}$, is given in equation (1.12), with the multi-phase optimisation problem detailed in equation (1.13).

$$\underset{x,y,z,\mu}{\text{minimise}}\ \bar{L_p}(x,y,z,\mu) = F_0(x) + \sum_{i=1}^{N} w_i.f_i(y_i) + \sum_{i=1}^{N}\{\mu_i(c_i(y)+z^2) + \frac{w}{2}|(c_i(y)+z^2)|^2\} \tag{1.12}$$

$$\begin{aligned} z^{k+1} &= arg\ \min_{z} |\bar{L_p}| \\ x^{k+1} &= arg\ \min_{x} \{\bar{L_p} : c_0(x) \leq 0\} \\ y^{k+1} &= arg\ \min_{y_i} \{f_i(y_i) : c_i(y_i) \leq 0 \in i = 1,...N\} \\ \mu^{k+1} &= \mu^k + \alpha.w(c_i(y)+z^2) \end{aligned} \tag{1.13}$$

[10] In such that any method iterating between a leader and follower design problem without any significant mathematical coupling can be considered a virtual Stackelberg game.

Where z represents some slack variable vector, creating effective equality constraints from inequality constraints $c_i(y_i)$ [121]. x represents a vector of system and linking variables $[x, x_0]$, it has been shown that dominant linking variables must be optimised as part of the aggregate top-level function [122]. y^k represents a vector of unique local solutions at iteration k, μ is the Lagrangian multiplier, updated by the outer method of multipliers process with respect to given weighting, w and outer step size, α.

Essentially ADMM creates an aggregate function, with penalties linked to local constraints with an iteratively updating Lagrangian multipliers for scaling variable coefficients. ADMM maintains primal consistency by approximating follower behaviour with a weighted penalty function. By restrcting linking variables to the leader-problem the result is relaxed follower feasibility, with increasing penalties for constraint violation with each subsequent phase. Where ADMM differs from the other methods presented is that it requires significantly less information transfer and coordination, much like ATC. It also and does not require the creation of artificial functions through reduced-order modelling or surrogate modelling like BLISS.

A general analysis of a numerical example shows that ADMM has excellent convergence properties under a range of state variables, w and α [123]. ADMM has been demonstrated on a coupled aerodynamic shape optimisation showing improved convergence performance and final solution when compared to a traditional gradient-based optimisation, for both 2D and 3D aerofoils [77].

1.5 Cooperative Behaviour Models

Game-theoretic approaches in MDO and engineering optimisation have become more common with the growing need for concurrent and collaborative product design [122, 124]. The primary drivers for said growth are the sharing of expertise, expanding solution spaces in chasing slimmer margins within evolutionary[11] product development, and risk/ redundancy mitigation in increasingly costly technological research [125].

In game-theoretic approaches, design subproblems are often referred to as decision-support problems (DSP). With each problem featuring, a set of partitioned decision parameters, each has agency over there own decisions, and each requires the support of one or more adjacent problems.

[11] Evolution with respect to product lift cycles; refers to the incremental improvement of over time as opposed to entirely new or revolutionary product development.

1.5.1 Game Theoretic Protocols

The main difference between collaboration protocols is in the direction and order when sharing information between each agent. Information handling is best exemplified using some of the more popular protocols within the literature: *Pareto*, *Nash* or *Stackelberg* protocols of cooperation [126].

Pareto Cooperative: Each agent has full access to information about each others decision and associated models. Each DSP has an exact representation of non-local decision parameters, or enough which is necessary to affect the local solution. In practise with collaborative inter-disciplinary problems this is impractical or difficult to implement due to poor compatibility of problem interfaces, additional lead time in sub-processes and the loss of control/ propriety knowledge. The resulting optimisation problem takes the form shown in equation (1.14), where f_0 and f_1 are two separable DSPs, with some cooperation multiplier α. Pareto cooperation is a common method used in multi-objective optimisation.

$$\underset{x,x_0,y}{\text{minimise}}\ \alpha.f_0(x,x_0)+(1-\alpha).f_1(x_0,y) \tag{1.14}$$

Nash non-cooperative: Agents cannot receive information from one another, with each making decisions independently of one another. Each DSP therefor must take rational local decisions assuming some best reply correspondence from an adjacent model. Response surface models are used to simulate best responses, approximating the multi-system solution space. If the resulting feasible solution sets overlap for any given set of variables a solution can be found to the overall problem.

$$\underset{x,x_0}{\text{minimise}}\ f_0(x,x_0) : x_0 \in \tilde{f_1}(x_0) \tag{1.15}$$

$$\underset{x_0,y}{\text{minimise}}\ f_1(x_0,y) : x_0 \in \tilde{f_0}(x_0) \tag{1.16}$$

The resulting optimisation problem statement is in equation (1.15), and (1.16). Here the linking variables, x_0, is a subset of the feasible solution space within some artificial function or surrogate, $\tilde{f_0}, \tilde{f_1}$.

Stackelberg leader/follower: Information and influence of domains is unidirectional, when a single agent is particularly dominant. The leader makes decisions first based on assumptions that the follower behaves rationally, with respect to some artificial function $\tilde{f_1}$. The follower then makes decisions without the need for assumptions about the leader.

$$\underset{x,x_0}{\text{minimise}}\ f_0(x,x_0) : x_0 \in \tilde{f_1}(x_0) \tag{1.17}$$

$$\underset{x_0,y}{\text{minimise}}\ f_1(x_0,y) : x_0 \in f_0(x_0) \tag{1.18}$$

The differences between the presented protocols and the optimisation architectures appear to be in the use of language, the scale of the problems presented and the practical obstacles of integration. Functionally the optimisation problem statements presented to mirror the effective methods used in MLO and MDO.

It is possible to equate the monolithic methods and cooperative protocols. The MLO engineering problem has shown a lot less uncertainty when it comes to sharing of propriety technology, information and expertise is much less of a design obstacle. However, the existing problems with monolithic methods still occur, even when dealing with fewer dimensions as in game theory.

Dual distributed methods broadly mirror non-cooperative protocols. However, due to the increased size of problems in MLO it is often computationally prohibitive to couple subproblems using a surrogate function of the feasible solution space. Similar to ATC, and ECO some outer loop process is used to ensure multi-criteria system feasibility and consistency of linking variables.

Whilst primal distributed methods match up with the unidirectional and sequential leader/follower protocols. As in the case in BLISS, the coupling is achieved through artificial functions representing lower-level system behaviour, or in ADMM where an aggregate leader function defines the possible solution space for local follower problems.

The cooperative, non-cooperative and Leader/follower protocols have been widely explored and tested for lower-order design models under a range of conditions. In the general case, much like MLO in a hierarchical design, the cooperative protocol is used and the benchmark case for testing new algorithms.

1.6 Review Findings

The field of optimisation has become significantly important in engineering and industrial design. In the literature there is a wide range of robust optimisation schemes available and depending on the given problem one can chose from either gradient-based algorithms, stochastic algorithms or hybrid architectures. Multilevel optimisation has added complexity onto a topic which is already broad and well actioned. Applications for MLO in design are far-reaching and often complex, resulting in an NP-hard classification of problems depending on the nature of the individual problems, as well as the relationship between resulting subproblems.

Hierarchical design problems posed as MLO problems are common within engineering research, with many studies, and cross-research possibilities with existing MDO methods. The significant distinctions in methodology come from the arrangement and interaction of subproblems with the major subsets presented, including monolithic solvers, primal distributed solvers, and dual distributed solvers.

1.6.1 Algorithm & Theory Summary

Monolithic solvers create a single optimisation problem, combining all variables, constraints and objectives into a single or multi-phase process. They are the simplest method to implement and does not need any special coordination schemes to account for interdependence or linking variables. A monolithic architecture results in optimisation problems that are often too large to be solved using globally optimal methods, relying on gradient-based solvers. Monolithic methods do not match the existing engineering design environment and may not be the most computationally efficient solutions for expensive multi-model simulations.

Distributed solvers, primal or dual methods, nested or alternating are disparate and numerous. These methods rely on iterative external coordination schemes to present a fully decomposed set of individual optimisation subproblems for each level/ model present in an analysis. Due to the range of research available, the difficulty associated, and the varied nature of particular research problems, there is little to no consensus on a generic modus operandi.

1.6.2 State of Research Summary

The breadth of multilevel architectures and research finds that merit or limitations depend solely on the problem formulation and model characteristics. There are few if any general benchmarking optimisation studies, making it difficult to accurately trade-off any one method over another.

Most engineering design studies presented test solutions against no more than one type of solver, using a traditional monolithic method as a datum or focus on too small a problem for practical application. Purely numerical or analytical applications of MLO very rarely discuss the practical aspects of problem parameterisation, or whether the system: subproblem used had an optimistic or pessimistic relationship. The result being when generating new or unique optimisers researchers may unintentionally bias the formulation to a bespoke problem. Generic frameworks overcome these problems such as MDAO and iSight by allowing for multiple solvers and multiple optimisation definitions of a single problem. [127–129]

A consensus on best practice is unlikely; instead, methods are applied on and a case case-by-case basis. Most engineering structural design examples of MLO found within the literature focus on simulation methodology, opting for simplified coupling methods. The result is a body of engineering research and industrial design methods which develops slower than some of the more promising methodologies in the field.

Whereas previous studies have focused primarily on the implementation of unique methodologies, this thesis focuses on the comparison of different existing algorithms available within MLO in terms of efficiency, accuracy and ease of use.

1.7 Chapter Summary

This chapter aimed to present an understanding of optimisation, MLO and streamline a broad range of literature. The landscape of work in optimisation and multilevel algorithms encompasses hundreds of disciplines, with implications far and exceeding the boundaries of engineering design. As such, the current state of the art on presented algorithms and methodology is just as broad without any final consensus on best practice. The work presented in this chapter has provided the following:

1. A formative background understanding of optimisation, problem statements and context defining a range of gradient, stochastic and hybrid algorithms.
2. Provided insight into the scope and sizeable applications of MLO in hierarchical engineering design and wider research, including network design and cooperative behavioural models.
3. Outlined a compilation of decomposition theory and multilevel algorithms, with the definition of terms: quasi-separable problems, optimistic, pessimistic and linking variables.
4. Explored the applications of MLO within engineering design, with reference to the major subsets of architecture: monolithic, primal and dual distribution.
5. Discussed the contribution of this thesis to ongoing optimisation research. Specifically, at building models based on a generic engineering framework, aimed at bringing MLO methods into the industrial environment.

The next chapter outlines the design problem, model development and the parameterisation process for a metallic wing model. These models are intended to mirror the current state of design and level fidelity used in an industrial context with the hope of providing an accurate blueprint of development for models in the future.

CHAPTER 2

MODELLING & DESIGN PARAMETERISATION

This chapter covers the development, decomposition and implementation of the models, objective & constraint functions used in this thesis. The system-level model is a procedurally generated FE wing-box structure, with responses and gradient information evaluated using MSC NASTRAN. The subsystem-level design models are an FE substructure replicating a stiffened upper-panel and a stiffened lower-panel man-hole. Each model generated was tested in isolation on a baseline rectangular wing-box model, considering aspects of scaling, normalisation and local convergence. Additional checks were considered for validation, scaling and parallel processing of design models.

The system and subsystem level models will use the layout shown in figure 2.1. The system-level is a global wing sizing problem based on a General FE Model (GFEM) featuring all significant wing components, including ribs and spars. The subsystem-level models will use a stiffened upper panel model, and lower panel model based on a Detailed FE panel Model (DFEM) for a single component.

Modelling techniques at each level of design will exhibit different behaviours or non-generic design features which would otherwise not have been present at the previous level. The sub-structuring regimes and techniques used will replicate the current best practice in industry and research. This investigation will focus on the methodology of coupling multi-model structural designs into a single architecture, rather than on the novelty of the specific modelling methodology [37, 102, 103, 115–120].

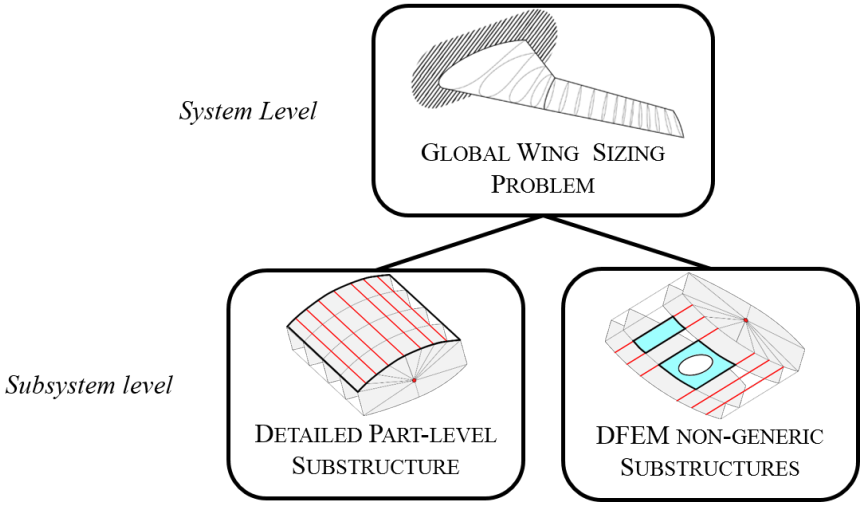


FIGURE 2.1. MLO Layout; Structural model sub-structuring regime uses a general wing model at the system-level, and local upper and lower skin panel models at the subsystem-level.

2.1 System Level Model

The General FE Model (GFEM) used in this work is a fully-parameterised, half-span model of a metallic wing. This GFEM will act as the system-level representation of the primary wing structure. The model has representative elements for all significant wing components, including ribs, spars, stiffened skins as well as the distribution of material properties. In order to ensure that the wing model and sub-models are consistent, the parametric representation of the GFEM was used to act as a generic repository for sub-models.

The GFEM uses a MATLAB® based wing model and data-set parameterisation. Using a modular design framework, the FE models and the optimisation were decoupled. The independence of each modelling component allows for concurrent 3D mesh generation, parallelised analysis, and independent checks of each model and analyses. The aim was to replicate the levels of detail available within industry, while creating a data repository which could be modified, duplicated or accessed by a range of optimisation routines.

The final parameterisation allows the variation of the following:

- FE Mesh Data, including mesh coordinates, metallic material properties, interfaces & property distribution.
- Wing Input and Geometry, including wing profile, span, wing inflection, internal structure and material definition.
- Skin thickness & sizing data for the upper skin, lower skin, ribs and spars.

The parametrisation allows for the variation of a range of external wing geometries and internal structures on a 3D half-wing FE model. Figure 2.2 shows the MATLAB-based representation and the equivalent FE model, with colours indicating the property distribution. The data generated is stored with loading, geometry, boundary conditions and mesh data structured to ensure consistency across multiple models and sub-models.

MSC-NASTRAN® is the primary solver, with `SOL 200` based sensitivity analyses [117]. Data processing and model evaluation have been separated, allowing each to be modified individually and for new FE models to be added without changing the model inputs or outputs. Local copies of model data can be made for N subsystems, ensuring models can exist independently and concurrently with one another.

Ribs and spars are idealised as 2D plate elements within NASTRAN. The half-wing is pinned at the root about the inner-most rib interfaces with the spars and skins. Skin cover sizing is parameterised using $2N$ equally spaced control points along the forward and aft edges (N points per edge), resulting in a span and chord-wise thickness variation for each sub-structure, plus an additional point at the tip ($2N+1$ control points in total). The thickness follows a 2D Lagrange distribution between adjacent control points [130].

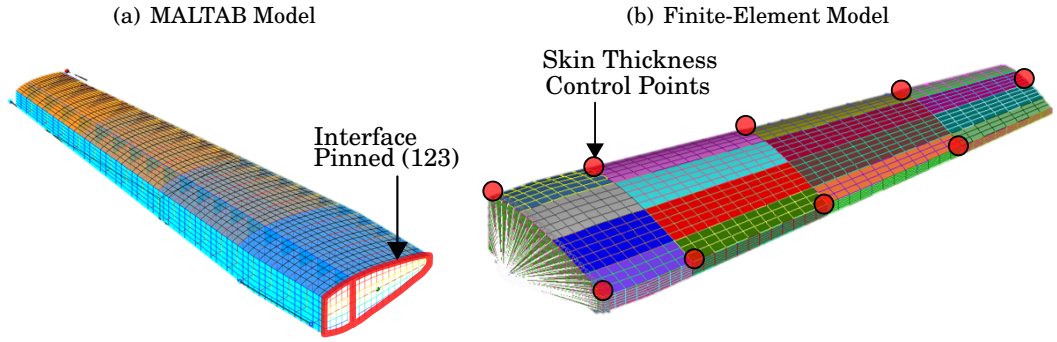


FIGURE 2.2. System-level Model; MATLAB 3D wing model and equivalent GFEM rendered in the FEMAP preprocessor for $N = 5$ section wing.

Stiffener modelling at the system-level uses an equivalent composite laminate to represent the smeared mass and stiffness properties of a unidirectional stiffened panel. The properties are calculated by solving for the equivalent skin thickness, density and material properties of an additional layer laminated onto the skin [131]. The material and geometric properties of the equivalent panel are then applied to the wing upper and lower skins as composite laminates, allowing for a low order modelling of the stiffened panel behaviour, and makes automated meshing and data transfer to substructures simpler.

For additional information on the modelling and parameterisation process see appendix 7.2.

2.1.1 System Level Model Summary

Model Inputs: $\boldsymbol{x}$, which represents rib, forward and rear spar thickness for N wing sections. The linking variables, $\boldsymbol{x}_0$, are the upper and lower skin thickness.

Model Outputs: Objective, F_0; Total wing mass, M; Constraints, including tip displacement, δ, stress, $\sigma_{1,\cdots,m}$, at m equally spaced control points (60 per wing section for the upper and lower skins); $\sigma_{1,\cdots,p}$ at p control points spread across rib and spar panels.

2.2 Subsystem Level Model

A Detailed FE panel Model (DFEM) will be used to model stiffened panel substructures along the upper and lower wing skin. The DFEM uses a greater mesh density and offset 1D beam elements to represent stiffeners. The DFEM is parameterised alongside the GFEM, with each panel removed and replaced by the more detailed substructure, as in figure 2.3. The parametrisation and meshing uses the same modelling framework shown in appendix 7.2.

An exemplar upper surface model is shown in figure 2.4(a). A lower surface model, including an inspection hole or man-hole plank (MHP), is shown in figure 2.4(b).

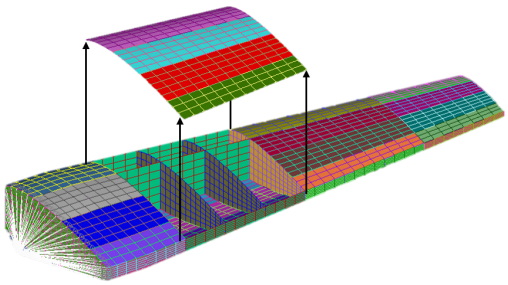


FIGURE 2.3. Panel Level model Sub-structuring; Equivalent panels are detailed replicas of existing panels removed from GFEM, with additional features.

(a) Upper Stiffened-Skin Model

(b) Lower Stiffened-MHP Model

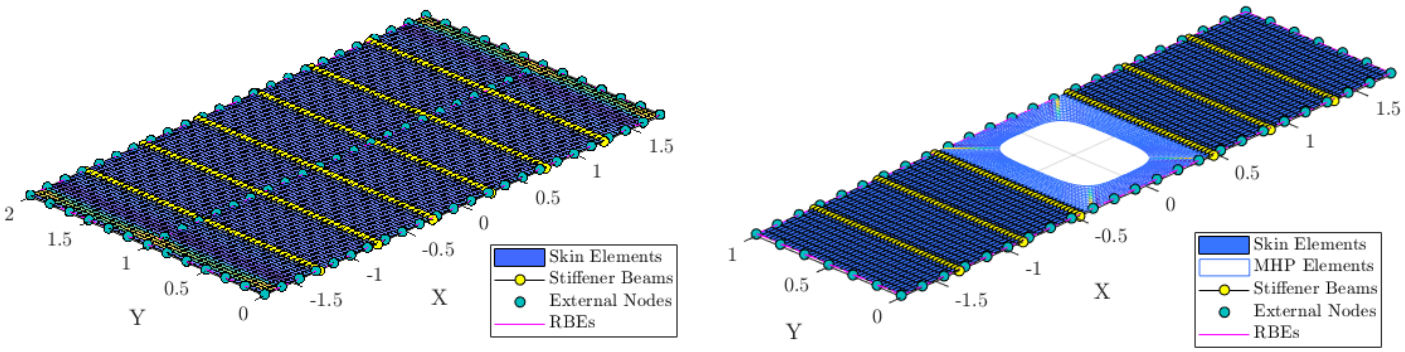


FIGURE 2.4. Subsystem-level Models; DFEM featuring 1D beam stiffeners, 2D skin and MHP elements, with external load introductory nodes attached using multi-point RBEs.

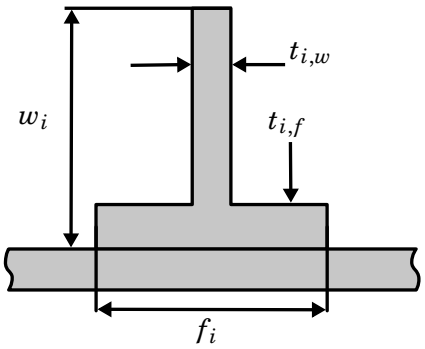


FIGURE 2.5. Stiffener Sizing Dimensions

The stiffener sizing dimensions are shown in figure 2.5 using the PBEAML sizing definition in MSC NASTRAN. Stiffener size indexing was initially considered to combine design and shape optimisation allowing for a greater range of stiffener types, i.e. I, C and box sections [116]. However, indexing could not be used in this study due to limitations within the NASTRAN based sensitivity analyses.

2.2.1 Substructure Panel Modelling

The *super-element* SE is combined with the local residual model stiffness matrices to accurately represent the external structure acting upon it using the LIST SE method in MSC NASTRAN. Each DFEM and SE model has been decoupled from one another to allow for the processes to occur concurrently between different panels. The buckling modes calculated using the SE are local to each panel, treating each as an independent model with stiffness, load and damping matrices. One must assume that substructure component buckling can be decoupled from one another [132]. Hence, resulting local modes are not coupled with global/adjacent buckling modes.

Expected stress peaks exist at load introductory points along rib: spar interfaces. As long as data is collected far-field of interfaces, gradient information for maximum stresses are meaningful and comparable across global and local models. Load redistribution in the DFEM due to stiffeners and greater model flexibility leads to a significantly lower average stress across the panel. This phenomenon was isolated and tested for a concurrent parallel architecture using a variable number of stiffeners and then bench-marked against equivalent analytical calculations to ensure robustness across the GFEM and DFEM.

For further information on the validation process of SE and multi-fidelity substructuring see appendix A.2.

2.2.2 Subsystem Level Model Summary

Model Inputs: $\boldsymbol{x}_0$, local skin thickness; $\boldsymbol{y_i}$, stiffener sizing and MHP circumferential thickness for the lower skin model.

Model Outputs: Local panel mass, M_i; maximum stresses in the stringer, $\sigma_{i,str}$; local buckling mode eigenvalue, λ_i, and stresses about a number of control points on the lower skin MHP, $\sigma_{i,MHP}$. All given relative to the i^{th} of N total sub-models. The DFEM subproblems have approximately 1500 elements, varying with the size of each section /bay.

2.3 Optimisation Design Summary

In summary, the design models use a metallic system-level GFEM of primary wing structures. Subsystem-level models are DFEM panels, i.e. substructures of the upper and lower wing skin exhibiting detailed behaviour and design features not present at the previous levels. These models use a MATLAB-based parameterisation ensuring consistency and between the different levels of design.

2.3.1 Baseline Rectangular Wing Model

The test configuration has a rectangular plan-form section with no taper or sweep. This simple design was chosen so sub-structure partitions lead to similarly sized subproblems in terms of both dimensions and computational requirements. This model can contain most of the features of the detailed features of a full wing-box model. This reference structures is also readily comparable to analytical solutions.

The wing-box has 16 ribs, with a chord of 3.5 m, maximum height 0.8 m and a span of 16 m. The wing-box is made of an isotropic, 2000-series aluminium alloy, with Young's modulus, E = 73.1 GPa, Poisson's ratio, $\nu = 0.33$, and density, $\rho = 2,900\,\text{kg/m}^3$ [133].

A stringer pitch of 0.3 m leads to four T-shaped stiffeners attached to the upper and lower skins. The box model generated is shown using the FEMAP pre-processor in figure 2.6, with different colours representing the property variation along the wing. The GFEM shown will act as the system-level model, with approximately 6000 shell elements.

The test model is fully fixed at the root, with a tip load of 100,000 N, applied at the centroid of the wing. The wing is split into $N = 8$ equally spaced 2 m sections defining substructure boundaries.

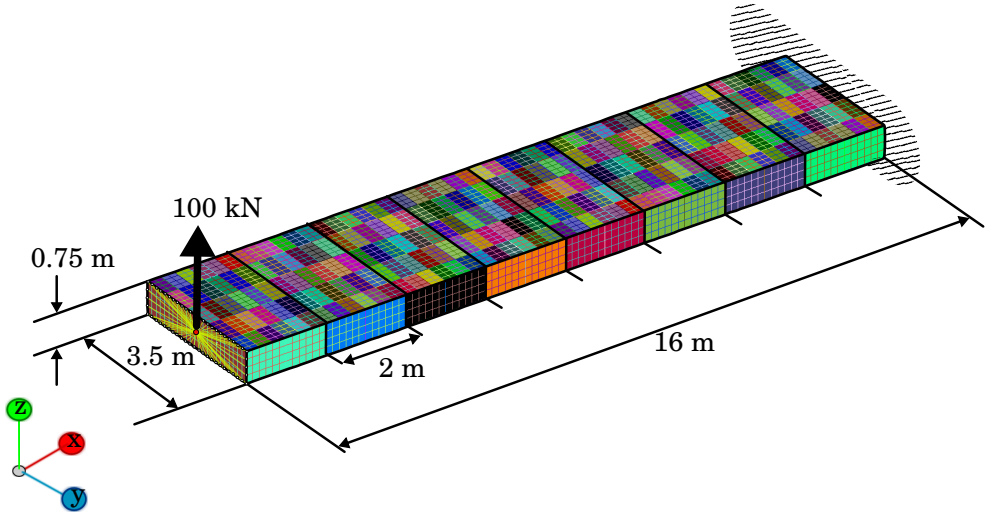
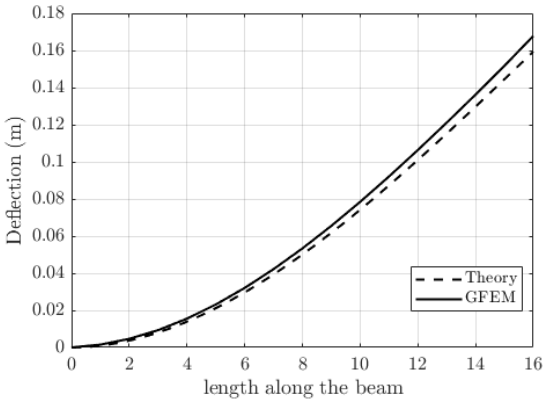


FIGURE 2.6. Baseline System Level Model; Rectangular Wing-Box FE Model

2.3.2 Model Validation

Procedurally generated FE models passed rigid body model checks under free-free conditions, with 6 elastic modes in the order 10^{-4}. Figure 2.7 shows a comparison of the deflection and stress distribution of GFEM/ DFEM models against simple 1D beam theory [134]. The system: subsystem parametrisation and modelling techniques show good agreement. The maximum discrepancy in measured stress away from loaded sections at root/tip is ≈ 4-8%, noting measured stress in a 3D box model using smeared stiffeners should have some discrepancy from the limited beam theory. The upper skin DFEM panels show an ≈ 1-2.9% variation from the maximum GFEM panel stress. The lower panel DFEM with the additional MHP feature results in an increase of 51.9% closer to the root, and just a 0.25% variation at the tip.

(a) Deflection



(b) Von-Mises Stress

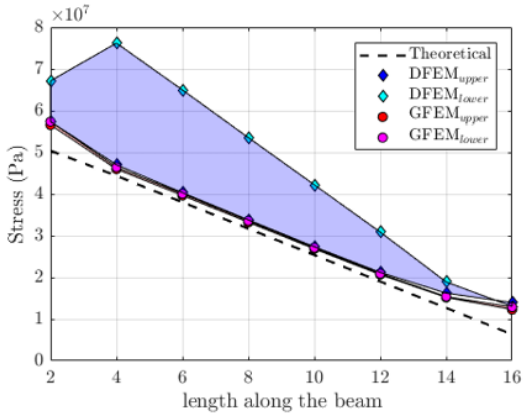


FIGURE 2.7. Deflection and Stress Comparison for System: Subsystem Models

2.3.3 System-Subsystem Design Synopsis

The coupled multi-model problem will be based on quasi-separable assumptions. The optimisation results presented are given for two separate design cases under different assumptions:

- *Optimistic Position*: Assumes the subproblems will produce results in favour/support the overall problem solution. These models will use the assumption that the measured stress from the GFEM system-level is sufficient for design. The DFEM subsystem-level will only consider local buckling and stringer.
- *Pessimistic Position*: Assumes subproblems solutions will worsen/differ from the overall problem solution. These models will use both the measured stress from the GFEM at the system-level, and the panel stresses from the DFEM subsystem-level. As shown in figure 2.7, the lower skin DFEM panels exhibit a significant increase in measured stress, presumably narrowing the overlapping feasible solution space between the different models.

The optimistic position uses assumptions to create a favourable relationship where local subproblems support the findings from the system-level problem. In practice, this achieved by assuming that the stress measured from the GFEM is sufficient. This assumption overlooks the local substructure stress, which includes significantly higher stress concentrations about additional features, including stiffeners and the MHP on the lower surface. By not including these constraints in terms of the optimisation problem the active constraints, those most expected to be infeasible and the objective function, shifts to being most likely to occur at the system-level. The result is that subproblem constraints are non-active but must remain feasible as part of the solution, and at no point in the design-space do they detract or limit the system-level objective.

The pessimistic assumption includes the limiting DFEM substructure stress constraints, shifting the active constraints to the subsystem-level. With potential active constraints at the subsystem-level, at some point in the design space, the subproblem will take priority and limit the objective function. Subproblems limiting the MLO design problem is overall, a more realistic expectation of detailed analyses, with the expectation being that only the most robust optimisation algorithms will adequately handle the more difficult of the two positions.

The sizing optimisation modifies skin thickness iteratively, minimising the structural mass, subject to strength and stability constraints. Panel thickness constraints based on design rules for structural joints, inserts or other fixtures, were not considered in this work. Objective and constraint function sensitivity with respect to design variables were evaluated using NASTRAN `SOL 200`, calculated internally using central finite difference for property variation [132].

A summary of the design variables used on the box model can be found in table 2.1. Overall, 130 design variables are employed, divided into 34 linking variables, 24 system-level variables and 72 variables at the local subsystem-level.

TABLE 2.1. Rectangular Wing-Box Design Variable Summary

Design Level	Description	Total
System Level, $\boldsymbol{x}$	Rib thickness	8
	Forward Spar thickness	8
	Rear Spar thickness	8
Linking Variables, $\boldsymbol{x}_0$	Upper Skin thickness	17
	Lower Skin thickness	17
Subsystem Level, $\boldsymbol{y}_{1-16}$	Upper Stiffener Dimensions	32
	Lower Stiffener Dimensions	32
	Manhole Plank Thickness	8

2.3.4 Constraint Screening

With reference to the fundamental MLO problem statement in equation (1.5), the design variables, $\boldsymbol{v}$, take the sizing and stiffener dimensions values, $[\boldsymbol{x}, \boldsymbol{x}_0, \boldsymbol{y}_{1,\cdots,16}]$, while the design objectives, $F_0 : f_i$, are the GFEM: DFEM mass.

Design variables and objective normalised have been normalised to improve the convergence properties of the optimisation problem. The design variables are normalised with respect to the bounded design range, where designs vary from $0 \rightarrow 1$, at the lower and upper bound, respectively. The objectives are scaled with respect to the sum of the combined Lagrangian form of the problem in order to improve convergence rate.

The constraints, c, too must be normalised adjusted to take the form $c(\boldsymbol{v}) \leq 0$. Design feasibility is then indicated by negative values and optimality by proximity to zero. Herein $c = \frac{c}{c_{\text{cr}}} - 1 \leq 0$, where critical datum's, c_{cr}, are given as follows:

- *Tip Deflection:* δ, measured at the central node on the last rib with respect to an arbitrary δ_{cr} = 15% of half span.
- *Crippling Stress:* Assuming $\sigma_0 = 68.9$ Mpa to be the minimum yield strength for 2000 series aluminium alloys [133], the crippling stress is calculated using equation (2.1) [118] with the following model responses:
  - Von-Mises stress along the upper and lower surface of the wing, $\sigma_{1,\cdots,m}$, is measured at m control points.
  - Von-Mises stress in the the spar/ribs, $\sigma_{1,\cdots,p}$ at p control points.
  - Von-Mises stress in local detailed panels, $\sigma_{i,1,\cdots,k}$ at k control points for substructures $i = 1, \cdots, N$.

$$\sigma_{cr} = 0.9 \cdot \sigma_0. \tag{2.1}$$

- *Spar/Ribs Panel Buckling Stress:* Using Von-Mises stress,$\sigma_{1,\cdots,p}$, on the spars/ ribs relative to an analytical measure of local panel buckling using (2.2); here K_L represents the panel dimension constant, t and b are the panel thickness and width respectively [135].

$$\sigma_{cr} = K_L . \frac{\pi^2 E}{12(1-\nu^2)} . \left(\frac{t}{b}\right)^2 \tag{2.2}$$

- *Local Stringer buckling:* Using Von-Mises stress, $\sigma_{i,str}$, on the local stringer blades, an analytical measure of stringer buckling can be calculated from the local DFEM. Assuming the local stringer blades are simply supported plates with one edge and equation (2.3); where E is material modulus, ts and d are the stringer blade thickness and width. Dimensions used were the flange width, $0.5 * f_i$, and web height, w_i, as shown in figure 2.5 [136].

$$\sigma_{cr} = 0.385 . E . \left(\frac{ts}{d}\right)^2 \tag{2.3}$$

- *Panel level Buckling mode:* λ_i with 1.

2.4 System Level Optimisation

In order to test model generation and the process of extracting gradients, a cover sizing problem was assessed using the GFEM. The extended design structure matrix (XDSM) [137] in figure 2.8 illustrates the basic data and process flow for this sample problem, with the design problem definition given in equation (2.4).

$$\begin{aligned} \underset{x,x_0-\mathbb{R}}{\text{minimise}} \quad & F_0(x,x_0) = M(x,x_0) \\ \text{subject to} \quad & \sigma_{\text{rf},1-m}(x,x_0) \leq 0, \\ & \delta_{\text{rf}}(x,x_0) \leq 0 \\ & x^{\text{L}}, x_0^{\text{L}} < x, x_0 < x^{\text{U}}, x_0^{\text{U}} \end{aligned} \tag{2.4}$$

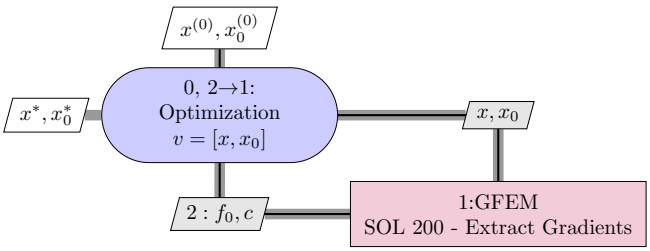


FIGURE 2.8. Upper Skin Cover Sizing XDSM

Figure 2.9 shows the iteration solution history obtained using MATLAB's `fmincon` function with default options. The default algorithm, *interior-point*, is an SQP-based solver which restricts step sizes between iteration to prioritise the feasible solution space. Prioritising feasibility and restricting step-size results in a different Lagrangian from the normal SQP constrained problem, reducing efficiency, but improves the reliability of the solutions [25].

The objective structural mass history in figure 2.9 is shown in *blue* on the left-hand y-axis, and the maximum active constraint violation (MaxC) history is shown in *orange* on the right-hand y-axis.

Figure 2.10(a) shows the system-level optimisation for multiple start positions, with respect to variables $\boldsymbol{v} = [x, x_0]$, using the standard SQP and the default interior-point within `fmincon`. A range of constraint handling options exist, the two shown here were the most efficient tested. All analyses converged to the same result, validating the model stability and convexity of the single-level design problem presented. Figure 2.10(b) shows mean design variable convergence for upper panel skin thickness between the upper and lower bounds, against the spanwise position. The repeatability of the model shows a converged solution variation of $\approx 0.01\%$ represented with the error bars on figure 2.10(b) barely visible at the scale shown.

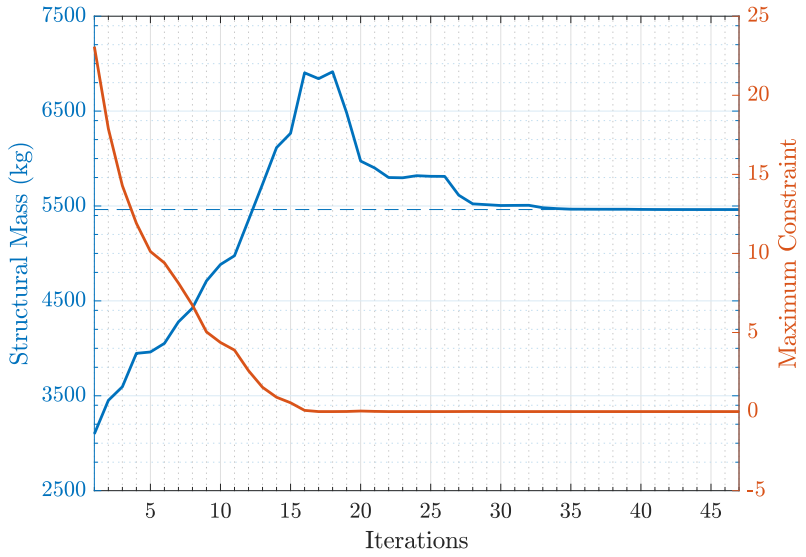


FIGURE 2.9. Sample Problem Optimisation Iteration History; Upper skin cover sizing problem convergence for multiple starting points and algorithms.

Variation in forward and aft skin thickness for a tip-loaded cantilever are due to the uneven distribution of m, n and p control points on the skins, ribs and spars respectively. These will remain constant for all analyses using the baseline model.

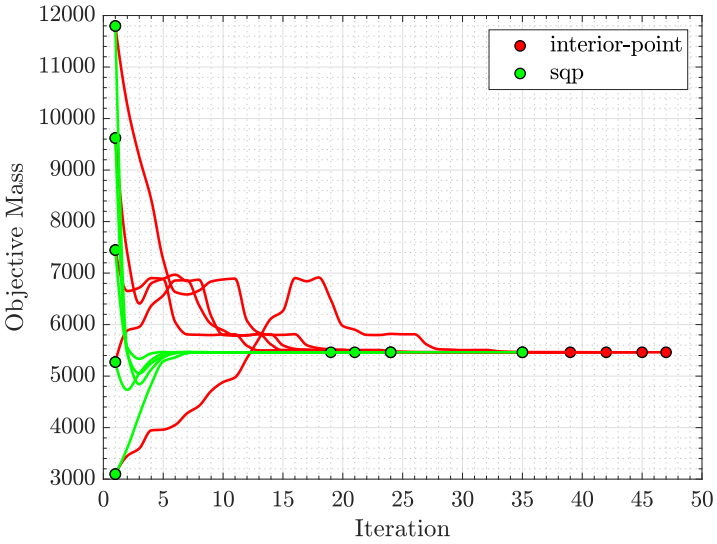
2.5 Subsystem Level Optimisation

The problem statement given in equation (2.5) represents the subsystem-level design optimisation. The selected methods must: (i) be efficient enough to find a solution with a reasonable number of function evaluations; (ii) consistently find satisfactory local optima while allowing for simultaneous and distributed parallel panel optimisation.

$$\begin{aligned} \underset{[x_0,y_i]-\mathbb{R}}{\text{minimise}} \quad & f_i(x_0,y_i) = M_i(x_0,y_i) \\ \text{subject to} \quad & \sigma_{\text{rf},1-k}(x_0,y_i) \leq 0, \\ & \lambda_{\text{rf}}(x_0,y_i) \leq 0 \\ & x_0^{\text{L}}, y_i^{\text{L}} < x_0, y_i < x_0^{\text{U}}, y_i^{\text{U}} \end{aligned} \tag{2.5}$$

The partitioned subsystem models must have sufficient analytical and behavioural separability to allow for each to be solved independently of one another. Accurate partitioning of variables and responses is indicated by the separability of system-subsystem problems, with weakly coupled distributed models [9]. Each panel DFEM can then be optimised separately and in parallel to improve the performance of the optimisation further. Parallelisation is a standard method in MLO and MDO to improve overall performance [34, 115, 138].

(a) Objective Function Iteration History



(b) Upper Skin Thickness Design Variable Convergence

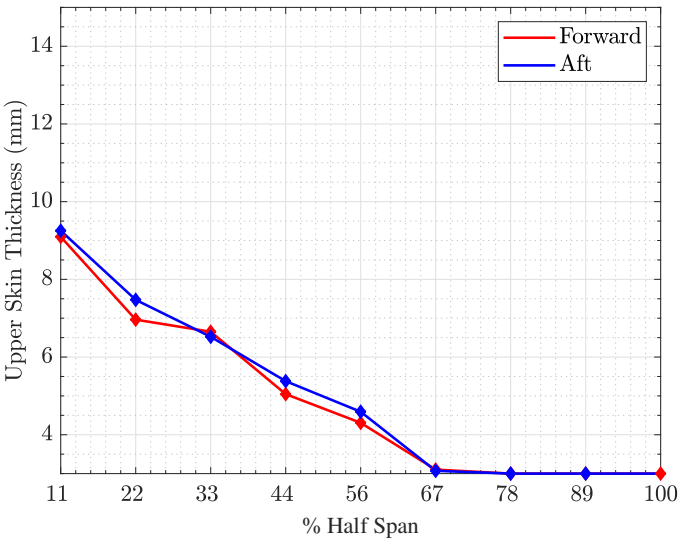


FIGURE 2.10. Sample Problem Convergence Distribution

2.5.1 Local Function Response

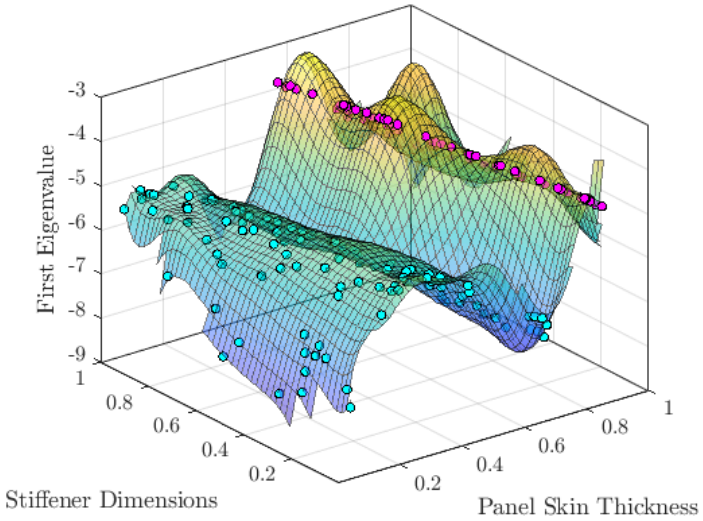
The selection of variables taken from table 2.1 are based on the physical sub-structures typically used in industry. Often for locally detailed models and stress consideration, there is limited scope for modification by independent teams outside of the immediate physical model.

Figure 2.11 shows the response surface models, in terms of panel buckling eigenvalue, for an upper skin DFEM buckling problem with respect to local skin-stiffener sizing variables [139]. Sizing variables are shown varying on a normalised scale from 0 to 1, with a univariate representation of local stiffener dimensions and panel skin thickness. The univariate response surface was used to assess the dominance of local variables on the subsystem-level problem and the robustness of the DFEM parametrisation. Local buckling behaviour shows significant sensitivity to the selected local sizing variables highlighting the potential structural reserves available by including local sizing variables into the initial concept sizing.

Figure 2.11(b) shows a top-down view of the buckling mode variation for a simple two-variable solution space. The design points shown in *blue* and *pink* represent the minimum buckling mode switching from one mode shape[1] to another. When design changes occur, eigenvalues drift, and mode switching can occur. Mode switching creates discontinuities resulting in a non-convex solution space and would present convergence difficulties for gradient-based algorithms. Note, the 'bumpiness' in the the response surface models are not representative on the true panel behaviour; they are numerical artefacts created in the kriging formulation, in part, due to the step change in the solution space.

[1] Mode shape is a specific pattern of vibration executed by a mechanical system at a specific frequency.

(a) Upper Skin Root Panel Buckling (3D view)



(b) Upper Skin Root Panel Buckling (Top view)

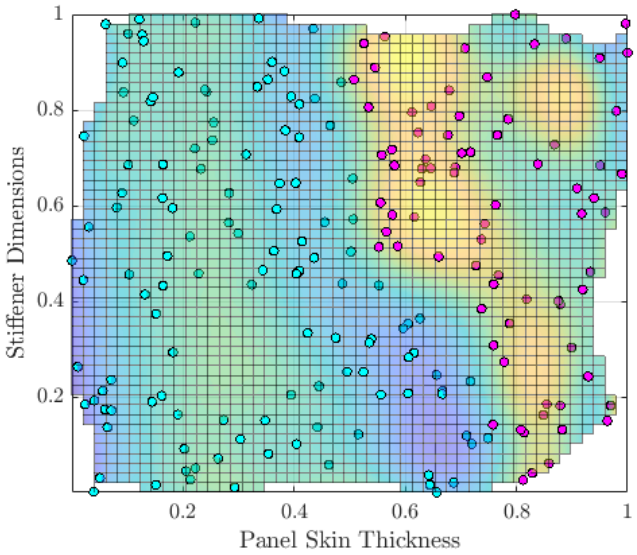


FIGURE 2.11. Response Surface Models for upper skin DFEM buckling behaviour with respect to local skin-stiffener sizing variables

2.5.2 Subsystem Constraint Screening

There are multiple methods available to deal with mode shifting, some of the most reliable found within the literature are: *mode tracking*, *constraint aggregation*, and using *globally optimal* optimisation methods. For further information on the implementation of these methods, see appendix B.2 on page 163.

Mode Tracking: Tracking eigenpairs between two subsequent analyses in order to ensure the same mode shape is observed throughout an analysis. Tracking is used whenever repeated buckling analyses take place to control discontinuities within the optimisation solution space. The tracking method testsed here will be the Modal Assurance Criterion (MAC). During the initial design step, the first 6 of 25 buckling modes are extracted and used as reference shapes for the subsequent analyses. Each step checks for the most similar buckling shapes in each consequent analyses, updating the reference each time [140].

Constraint Aggregation: Taking a large number of buckling modes at each design step, enough that would be sufficient for the entire spectrum of system behaviour to be captured, then combining them into a single value. The resulting aggregate reduces the computational costs of sensitivity analyses for large-scale solvers by combining a range of responses into a single value.

An adaptive Kreisselmeier-Steinhauser (KS) function will be used to aggregate constraints. The KS function converts a vector of design responses into a single value by approximating a smooth and uninterrupted solution space made up of the most significant response at any given point. The KS function uses a nominal 'draw-down' aggregation factor, ρ_c, and an aggregation sensitivity tolerance, KS'_d to control the smoothness of the resulting response surface [61, 141].

The first 25 buckling modes of the DFEM were taken with each design step, local responses were then aggregated (including crippling stress, and local stringer constraints). The KS function used nominal values $\rho_c = 50$ and $KS'_d = 10^{-5}$. In preliminary tests and as stated within the literature these are the most reliable state paramaters for this type of study.

Globally Optimal Optimisation: The benefit of distributed optimisation methods is that it allows for one to use different algorithms at different potions of the design. Partitioning constraints into optimisation subproblems allows the designer to select the most appropriate algorithm based on said subproblem. Stochastic methods, although inefficient for higher dimensional design spaces, do not encounter the same difficulties in discontinuous design spaces as gradient-based methods. However, these methods are only viable as long as the subsystem-level problem operates with a reduced number of constraints and design variables.

2.5.3 Surrogate Based Approach

The subsystem optimisation presented has a non-convex solution space, the final algorithm must be a globally optimal algorithm. For the given problem size, one can improve search efficiency through the use of surrogate models, with stochastic optimisers. The process used here is losely based on the EGO method and will be labelled as such accordingly. The full method is illustrated in figure 2.12 and broadly involves the following steps [38]:

Step 1: A Latin hyper-cube sampling (LHS) of the design space defines the initial design points x_p.
Step 2: The surrogate is created using a gradient enhanced kriging model [142], with a surface minima sampling radial basis function as the infill criterion [139].
Step 3: The root mean squared errors (RSME) of new design points are tested against surface minima. An RSME below a set tolerance indicates the surrogate can be trusted.
Step 4: The potential optima $[x^*]_p$, are calculated using the global search algorithm on the surrogate, $\tilde{f}(x)$.
Step 5: The final design step is a gradient-based step taken from potential optima on the true function.

Aggregated local constraints were used to reduce the number of surrogate responses required using the KS method, using notation $\bar{c_i}$ as seen in equation 2.6. The RSME is calculated across a range of design points periodically sampled throughout the optimisation process using a combination of a LHS and surface minima infill criterion. The resulting vector of aggregated responses, $[\bar{\boldsymbol{c}}]$, and the surrogate responses, $\tilde{\boldsymbol{c}}$, are then processed using equation 2.7. The EGO algorithm was tested and tuned using a series of standard numerical test functions; Ackley's, Rosenbrock and Six-min in both constrained and unconstrained forms.

$$\bar{c_i}(\boldsymbol{x}_0, \boldsymbol{y}_i) = KS_c([\lambda_{\mathrm{rf},i}, \sigma_{\mathrm{rf},istr}, \sigma_{\mathrm{rf},iMHP}], \rho_c, KS'_d)) \tag{2.6}$$

$$\mathrm{RSME}(\bar{\boldsymbol{c}}) = \frac{\mathrm{mean}(\sqrt{\sum(\bar{\boldsymbol{c}} - \tilde{\boldsymbol{c}})^2})}{\mathrm{mean}(\bar{\boldsymbol{c}})} \tag{2.7}$$

2.5.4 Algorithm Selection

For the subsystem optimisation problem presented a range of stochastic and globally optimal algorithms were tested. Table 2.2 shows a summary of the final results of each algorithm solving the upper-tip panel optimisation problem presented in equation (2.5). Each method used was based on the MATLAB global optimisation toolbox [25]. The multi-start algorithms were both given 10 uniformly distributed starting positions. The PSO 'nonclon' used a version of PSO capable of optimising for non-linear constraints [18].

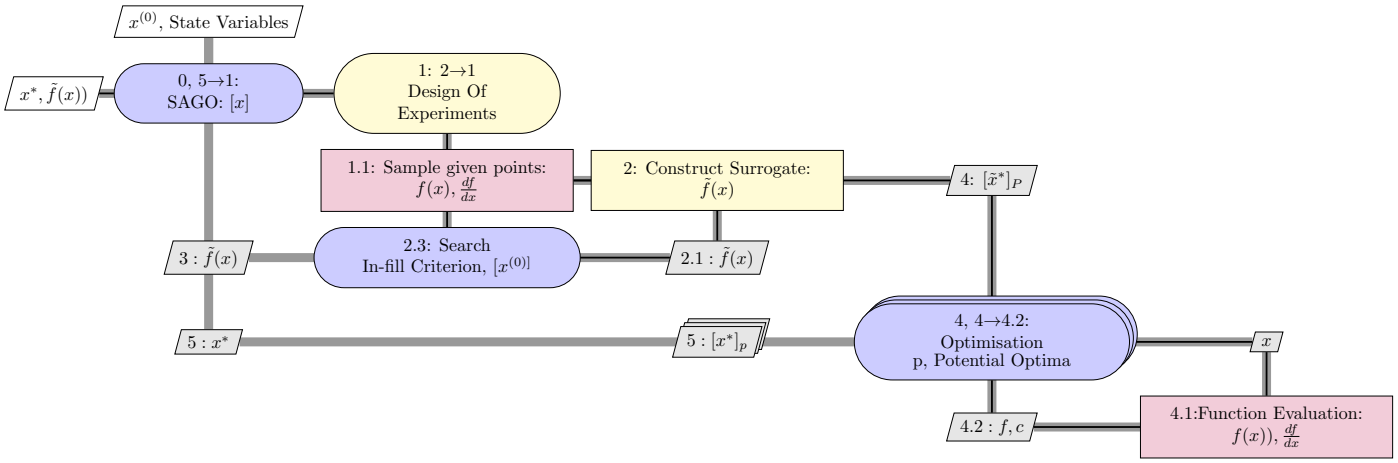


FIGURE 2.12. Efficient Global Optimisation Process XDSM

Note, the NASTRAN evaluations for the EGO method includes those used in the initial sample to build the surrogate. The surrogate evaluations are the number of evaluations of the artificial surrogate function; these are of relatively little to no computational cost in comparison to a single NASTRAN evaluation.

TABLE 2.2. Local Panel Buckling Optimisation Results Summary

Algorithm	Multi-start KS	Multi-start MAC	PSO nonclon†	GA†	Global Search	EGO*
Objective (kg)	79.358	84.391	75.606	88.094	83.253	74.691
Max Constraint	0.00	0.00	0.00	0.00	0.00	0.00
F-Evaluations	501	1084	1798	1430	3346	127 (5505)

† = Hybridised with a final gradient-based optimisation step.
* = NASTRAN Function Evaluations (Surrogate Evaluations in brackets).

All of the presented methods were able to accurately solve the presented problem, to varying degrees of success. The most efficient method used was EGO, with significantly fewer function evaluations within NASTRAN; however, there is an increased computational overhead in generating surrogates for the global optimisation step (these are included and make up the majority of the total evaluations shown). The best optimal panel design is shown in figure 2.13, for this simple panel problem a linear variation in thickness span-wise, and little to no chord-wise variation under the symmetric loading conditions. The thickness for webs, flanges, ts_{17} and flange width were all set to the lower bound value for this final design.

The global search algorithm used a reduced set of stopping conditions in order to reduce the number of function evaluations before convergence, resulting in a less competitive solution. EGO

used the global search algorithm to help construct the surrogate and define potential optima.

PSO and GA were both hybridised to provide a final gradient-based step to tune the final solution. These methods are population-based approaches which introduce random elements into the design, and both are susceptible to tuning state-parameters to support the problem formulation. Variance in stopping conditions, population size and mutation process could lead to significant improvements in the obtained solutions but these were not tested here [30].

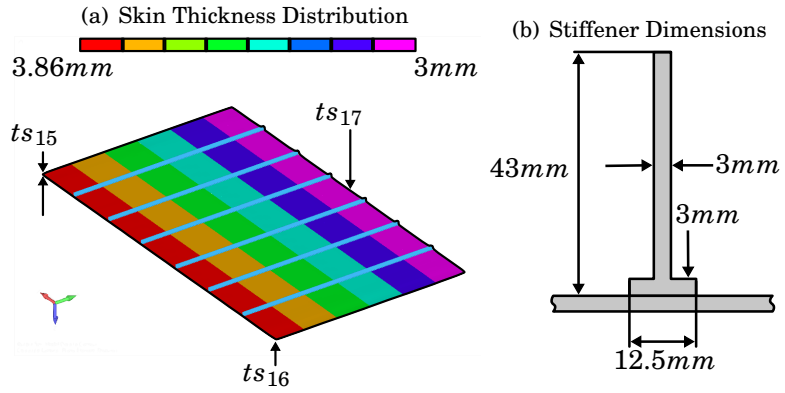


FIGURE 2.13. Optimal design for Local Upper Tip Cover Panel; converged solution from local subproblem optimisaiton

2.5.5 Distributed Concurrent Optimisation

The selected algorithm must apply to a variety of design spaces, with each subsystem presenting a different solution space. One of the requirements of these analyses was to allow for parallel design architecture. The Multi-start KS optimisation and EGO were tested to see how each method handles parallelised analyses for a selection of substructures.

Table 2.3 and 2.4 show the results of a concurrent upper skin buckling design optimisation, using the EGO method, and a multi-start method with 5 uniformly distributed start points (reduced from 10 previously). Figure 2.14 shows a comparison of the objective mass and function evaluations for both algorithms.

TABLE 2.3. Parallelised EGO Results Breakdown

Wing Section, i	1	2	3	4	5	6	7	8
Objective (kg)	200.887	181.762	154.748	149.229	130.408	113.590	98.723	74.786
Max Constraint	-0.019	-0.028	-0.014	-0.111	-0.048	-0.007	-0.073	0.000
F-Evaluations	681	477	608	608	817	592	632	198
S-Evaluations	176522	99761	201520	198522	103394	29645	33795	6071

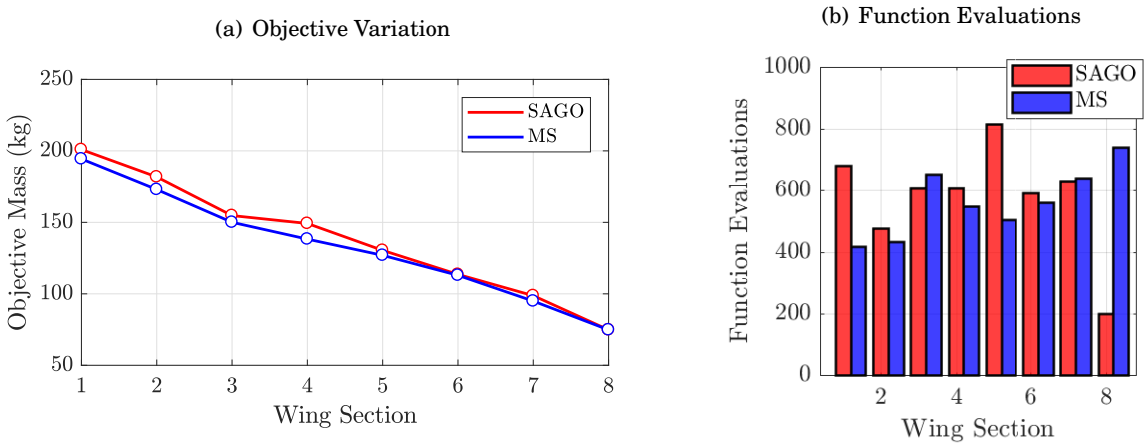


FIGURE 2.14. Comparison of Algorithms for a Parallelised Panel Upper Skin Panel Analysis

TABLE 2.4. Parallelised Multi-start Results Breakdown

Wing Section, i	1	2	3	4	5	6	7	8
Objective (kg)	194.335	172.978	150.042	138.276	126.930	112.948	94.923	74.769
Max Constraint	0.001	0.001	0.000	-0.003	0.002	0.000	-0.002	0.000
F-Evaluations	417	433	655	547	506	560	642	740

Both methods converge to a region close to the global optima, with the multi-start performing marginally better across most panels, except for the upper-tip panel, where the number of function evaluations is 3 times higher than the EGO method. The total NASTRAN evaluations was 4613 using EGO, and 4500 for the multi-start method. The state-parameters (e.g. sample size, infill criterion, error tolerances) used to construct surrogates were tuned using the upper-tip panel optimisation problem, or wing section $i = 8$ in the parallelised analyses. The result being that EGO only significantly outperformed the multi-start method for the tip panel, $i = 8$. For 'off-design' functions with different solution spaces, such as section 1 and 5, the multi-start method performed significantly better than EGO.

In the case of high-dimensional models the most efficient methods use a two-level process to generate surrogates. First optimising using EGO for surrogate based optimisation; with a second outer process optimising for state-parameters. By iterating between these processes one can gradually improve the quality of the inner optimisation solution and the resulting surrogate [143]. However this approach would need to be applied to each panel, the significant computational costs would have made any MLO method using EGO for subproblem optimisation prohibitively expensive.

The multi-start method is simple, robust and has the generality to be applied to a range of optimisation problems with adequately consistent performance. The dual-distributed procedure will use the multi-start method for local subproblem optimisation. The EGO method is reliable enough to build and train surrogates to find a globally optimal solution. If local surrogates are necessary for the analyses, the EGO method will be used for local subproblem optimisation, with a procedural update process in order to improve efficiency.

2.6 Chapter Summary

The chapter has presented the system and subsystem models to be used as part of the multilevel design problem. The presented models considered:

1. A system-level GFEM encompassing all of the significant wing components and parameterised with upper skin, lower skin, spar and rib thickness sizing variables. The system-level method uses MATLAB's `fmincon` algorithm, with function and gradient evaluation taken from the GFEM using NASTRAN `SOL 200`.

2. A subsystem-level DFEM consisting of substructures for the upper skin cover panel, and lower skin MHP bay, parameterised with thickness sizing and stiffener shape variables. The external loads and edge conditions defined using a NASTRAN based SE model. The subsystem-level model will use a multi-start optimisation for dual decomposed methods and EGO for primal decomposed methods requiring surrogates.

3. Design variables have been normalised on a scale of 0-to-1. Generic constraint screening is used for a series of failure modes, including buckling, stress and deflection, all with a greater than 0 value indicating feasibility.

4. Outlined the baseline MLO design problem used in the following chapter. Assumptions made outlined an optimistic and pessimistic design case.

   – The optimistic position has subproblems with less competitive constraints. These use the assumption that the measured stress from the GFEM system-level is sufficient for design. The DFEM subsystem-level will only consider local buckling and stringer.

   – The pessimistic positions produces more competitive subproblems with more challeging constraints. These models will use both the measured stress from the GFEM at the system-level and the panel stresses from the DFEM subsystem-level featuring higher stressed panel models.

The next chapters presents the system and subsystem level models integrated into a range of optimisation architectures. Each architecture is based on the current best-practise for a representative subset of the MLO literature.

CHAPTER 3

MONOLITHIC MULTILEVEL OPTIMISATION

The next chapters will detail the methods, processes and optimisation results for a range of multilevel optimisation architectures. This chapter details the monolithic single-phase and multi-phase optimisers, tested under a range of conditions and assumptions for comparison. The objective is to minimise the structural mass of a simple wing-box structure, with respect to system-level constraints and subsystem panel substructure constraints.

Monolithic architectures are the simplest method of multilevel optimisation to implement and is often used as the baseline for testing more novel solutions. When design variables and responses all exist within the same domain, in this real & continuous, iterating system and subsystem level models can be combined into a single step with local objective functions, f_i, and concatenating individual model constraints, c_i. Structural models will then be integrated into a single-phase and multi-phase monolithic multilevel solvers.

3.1 Monolithic Optimisation Methods

The combined monolithic optimisation problem statement is given in equation (3.1). Single-phase solvers attempt to find a solution to the problem in a single optimisation problem, while multi-phase solvers incrementally increase the level of computation and dimensions of the problem. The most common way to solve a multi-model problem with a combined gradient evaluation step is using *classical constraint decomposition*.

$$\begin{aligned}
\underset{\boldsymbol{v}}{\text{minimise}} \quad & F(\boldsymbol{v}) = M(\boldsymbol{x},\boldsymbol{x}_0) + \sum_{i=1}^{N} M_i(\boldsymbol{x}_0,\boldsymbol{y}_i) \quad \text{where} \quad [\boldsymbol{v}] = [\boldsymbol{x},\boldsymbol{x_0},\boldsymbol{y_{1\cdots N}}] \\
\text{subject to} \quad & \sigma_{\text{rf},1-m}(\boldsymbol{x},\boldsymbol{x}_0) \le 0, \\
& \sigma_{\text{rf},1-p}(\boldsymbol{x},\boldsymbol{x}_0) \le 0, \\
& \delta_{\text{rf}}(\boldsymbol{x},\boldsymbol{x}_0) \le 0, \\
& \lambda_{\text{rf},i}(\boldsymbol{x}_0,\boldsymbol{y}_i) \le 0, \\
& \sigma_{\text{rf},istr}(\boldsymbol{x}_0,\boldsymbol{y}_i) \le 0, \\
& \sigma_{\text{rf},iMHP}(\boldsymbol{x}_0,\boldsymbol{y}_i) \le 0, \quad \text{where section: } i = 1,\cdots,N \\
& v^{\text{L}} < \boldsymbol{v} < v^{\text{U}}
\end{aligned} \tag{3.1}$$

Classic Constraint Decomposition: Monolithic methods require local subproblem information to be combined with the greater system-level problem. Equation (3.2) shows the resulting constraint gradient matrix for the system-level problem with respect to design variable vector, $\boldsymbol{v}$ [91].

$$\frac{d\boldsymbol{c}}{d\boldsymbol{v}} = \begin{bmatrix} \frac{dc_0}{dx} & \frac{dc_0}{dx_0} & \cancel{\frac{dc_0}{dy_i}} \\ \cancel{\frac{dc_i}{dx}} & \frac{dc_i}{dx_0} & \frac{dc_i}{dy_i} \end{bmatrix} \tag{3.2}$$

The cross-boundary sensitivities of local responses with respect to linking variables, $\frac{dc_i}{dx_0}$, are extracted as part of the SE process in NASTRAN `SOL 200`. Effectively the substructure response is coupled with sensitivities from the external SE, for more information on this process see appendix B.1.

An assumption of classic constraint decomposition is that the sensitivities $\frac{dc_0}{dy_i}$ and $\frac{dc_i}{dx}$ are assumed null, with reasonably separable global: local behaviours. Assessing these sensitives using a combination of finite-differencing and coupled sensitivities within NASTRAN both derivatives have been evaluated, and have no appreciable effect on the overall system, i.e. 10 orders of magnitude smaller than those of local or linking variables.

3.1.1 Monolithic Single-Phase Optimisation Process Description

Algorithm 1 presents the AAO architecture with the programming description split into three main steps: system-level analyses, subsystem-level analyses and gradient formation. Variables with the superscript '*' or '(0)' indicates the value at its optimum and initial position, respectively, as listed in table 2.1.

Algorithm 1 AAO Optimisation Procedure

Inputs Starting position: $[v^{(0)}]$, Bounds: v^{L}, v^{U}
function FMINCON($v^{(0)}$, v^{L}, v^{U});
    Convert $[v]$ to model sizing x, x_0 and, local panel sizing $y_{1,\cdots,N}$;
    Create Equivalent laminate skin properties;
    System model: run GFEM `SOL 200` $\models F(x,x_0)$ and $c_0(x,x_0)$; ▷ Step 1
    parfor $i = 1 \rightarrow N$ **do**
        Subsystem model: run DFEM `SOL 200` $\models f_i(x_0,y_i)$ and $c_i(x_0,y_i)$ ▷ Step 2
    end parfor
    Concatenate Constraint and Gradient Matrices: $[c_0(x,x_0), c_{1\cdots N}(x_0,y_{1,\cdots,N})]$; ▷ Step 3
return Converged solution: $[v^*]$

The full AAO process description is in the XDSM of figure 3.1, the processing and data blocks are numbered to match the steps outlined in algorithm 1. Run-time variables of P_i and K_i, represent displacement and stiffness information from the GFEM super-element model to the DFEM residual model, recalculated for each iteration. R_i represents residual derivative information with respect to system and linking variables.

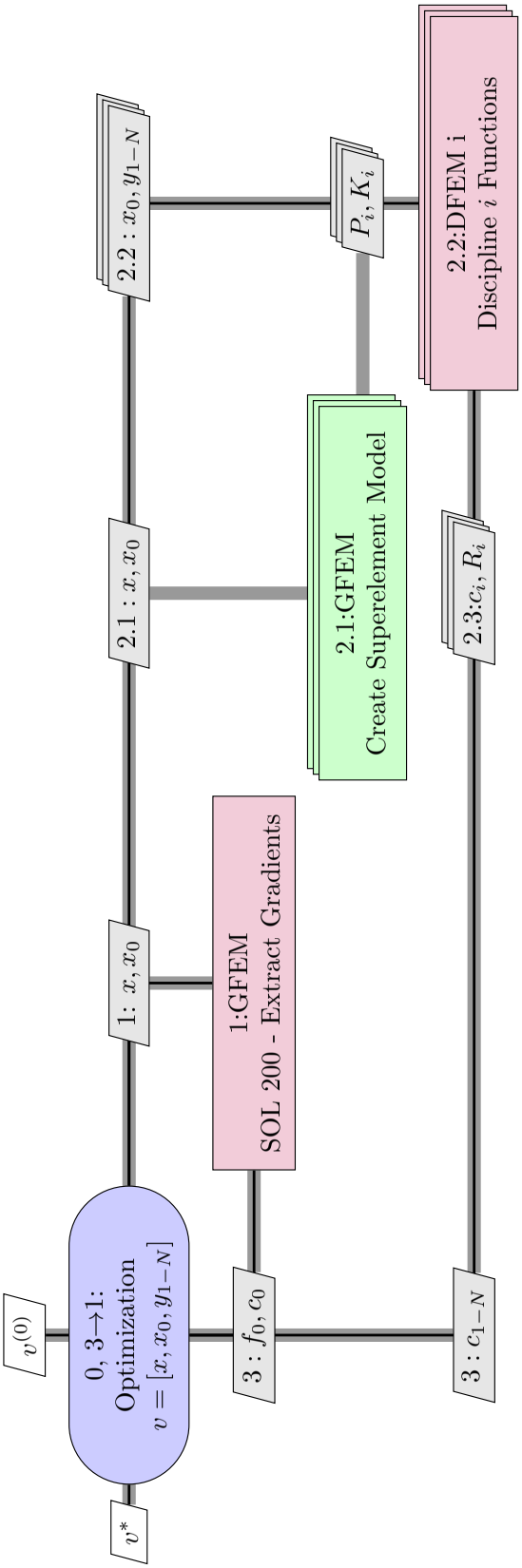


FIGURE 3.1. Diagram for the AAO procedure

3.1.2 Monolithic Multi-phase Optimisation Process Description

This scheme is based on the sequential level mixing process and composes of three levels; *level 1*, only using the GFEM system-level model; *level 2*, using both the GFEM and upper surface subsystem models; and *level 3*, using the GFEM, with all upper and lower surface models. Table 3.1, shows how the optimisation problem changes throughout with the number of design variables, constraints and function evaluations per design step. Each level includes the design variables and constraints from the previous one, retaining the coupled system behaviour. Algorithm 2 details the overall program description.

TABLE 3.1. Sequential Optimisation Levels Summary

Level Definitions:	level 1	level 2	level 3
Design Variables	58	90	130
Constraints	1233	1337	1537
Function Evaluations	1	9	17

Algorithm 2 Sequential Optimisation Procedure

Inputs Starting position: $[v^{(0)}]$, Bounds: v^{L}, v^{U}
Convert v to model sizing x, x_0 and, local panel sizing $y_{1,\cdots,N}$;
function FMINCON($[x^{(0)}, x_0^{(0)}]$, $[x^{\mathrm{L}}, x_0^{\mathrm{L}}]$, $[x^{\mathrm{U}}, x_0^{\mathrm{U}}]$); ▷ Level 1
    System model: run GFEM `SOL 200` $\models F(x, x_0)$ and $c_0(x, x_0)$;
Return Partial solution: $[x^*, x_0^*]$;
Update: Starting Positions $[x^{(0)}, x_0^{(0)}] = [x^*, x_0^*]$;
for Level, $L = 2, 3$ **do**
    Define N_L number of subsystem models at level L;
    Set level starting position and bounds: $v^{(0)} = [x^{(0)}, x_0^{(0)}, y_{1\cdots N_L}^{(0)}]$;
    function FMINCON($v^{(0)}$, v^{L}, v^{U}); ▷ Levels 2 and 3
        Create Equivalent laminate skin properties;
        System model: run GFEM `SOL 200` $\models F(x, x_0)$ and $c_0(x, x_0)$;
        parfor $i = 1 \rightarrow N_L$ **do**
            run DFEM `SOL 200` $\models f_i(x_0, y_i)$ and $c_i(x_0, y_i)$
        end parfor
        Concatenate Constraint and Gradient Matrices: $[c_0(x, x_0), c_{1\cdots N}(x_0, y_{1,\cdots,N_L})]$;
    return Partial Solution: $[x^*, x_0^*, y_{1-L}^*]$;
    Update: Starting Conditions $[x^{(0)}, x_0^{(0)}, y_{1,\cdots,N_L}^{(0)}] = [x^*, x_0^*, y_{1,\cdots,N_L}^*]$;
return Converged solution: $[v^*] = [x^*, x_0^* y_{1,\cdots,N}^*]$

3.2 Single-Phase Optimisation Results

Practical applications of optimisation in engineering design problems has meant that typical convergence is when the residual of the objective function is smaller than a given tolerance and satisfies all given design constraints.

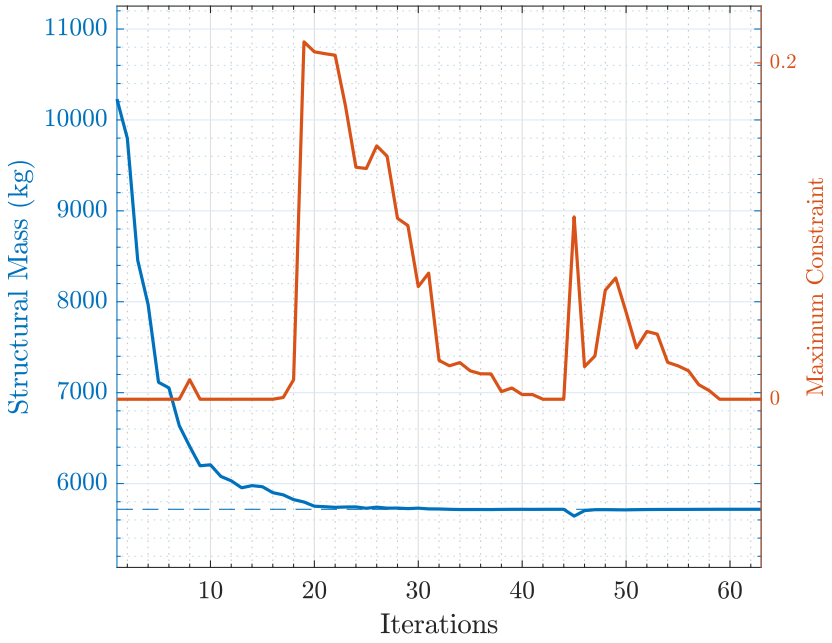


FIGURE 3.2. AAO: Optimistic Problem Function Iteration History; Satisfactory less than zero maximum constraint and final optimised mass, M^* = 5717.2 kg.

Figure 3.2 shows the objective and constraint iteration histories for the AAO optimistic design problem. Convergence for models was satisfied when the minimum step change in design variables would no longer produce any appreciable changes in design solution. In order to test the effects of the optimistic and pessimistic design assumptions on solution space convexity and constraint satisfaction, both assumptions were tested from a series of starting positions.

Whether using optimistic or pessimistic assumptions results in a local design problem with more or less conservative design space, respectively. The practical result of conservative assumptions in subproblems is that the active constraints shift from system-level to subsystem-level problems; resulting in the overlap between feasible solution spaces at the system and subsystem levels becomes narrower, increasing the difficulty of the design problem.

Figure 3.3 shows the normalised objective function as a percentage of optimised mass,M^*, given in figure 3.2. Points marked in *white* indicate convergence to an infeasible design, a greater than zero active constraint. Maximum constraint violation (MaxC) histories are shown in figure 3.4.

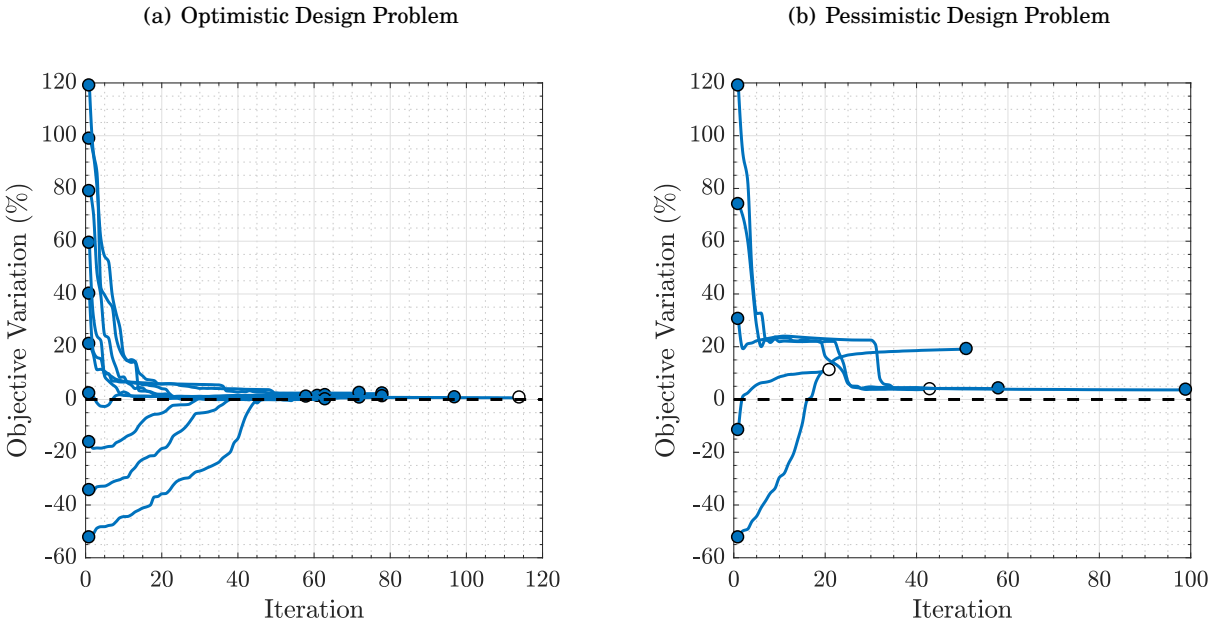


FIGURE 3.3. AAO: Multiple Start Objective Function Iteration History; Normalised in terms of the percentage objective variation from initial optimised mass M^*.

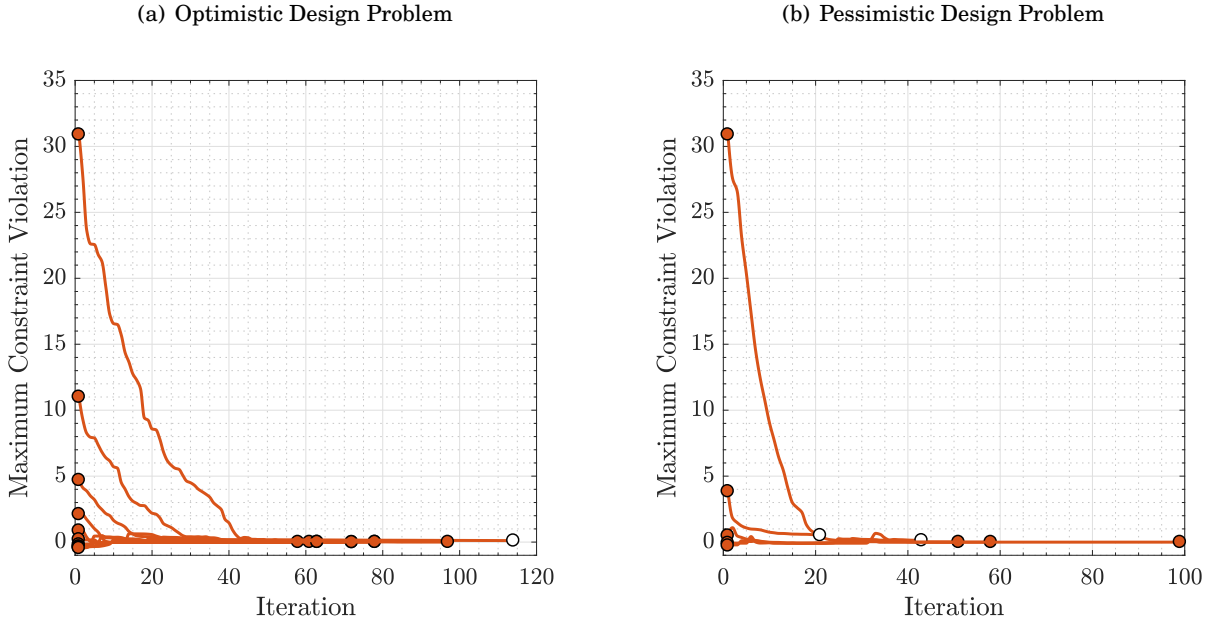


FIGURE 3.4. AAO: Multiple Start MaxC Iteration History; MaxC per iteration, points with white end markers have a greater than zero final constraint.

The maximum objective variation between converged solutions for the optimistic problem is 2.2% or 125 kg, compared to 19.1% or 1089 kg in the pessimistic problem. Using optimistic design assumptions the maximum error was 0.1038 appearing in 1/10 solutions; the pessimistic assumptions resulted in a maximum error of 0.5118, appearing in 2/5 solutions.

The introduction of additional design constraints at the local level reduces the reliability of the AAO monolithic method. The changes in AAO solutions under pessimistic conditions is consistent within the MLO literature supporting distributed architectures. Introducing conservative assumptions at different levels of design can have a significant effect on the viability of any of the methods presented.

For the majority of tests, the AAO method with optimistic assumptions leads to a reasonably consistent overall design. While using pessimistic assumptions, repeated tests were required for a viable solution. In the pessimistic case, it cannot be confirmed whether this is the true optima, as all solutions shown converged for minimum step change tolerance, rather than fully satisfying KKT conditions. For the practical use in engineering design problems, the starting position using monolithic methods would need to be reasonably close to the expected design.

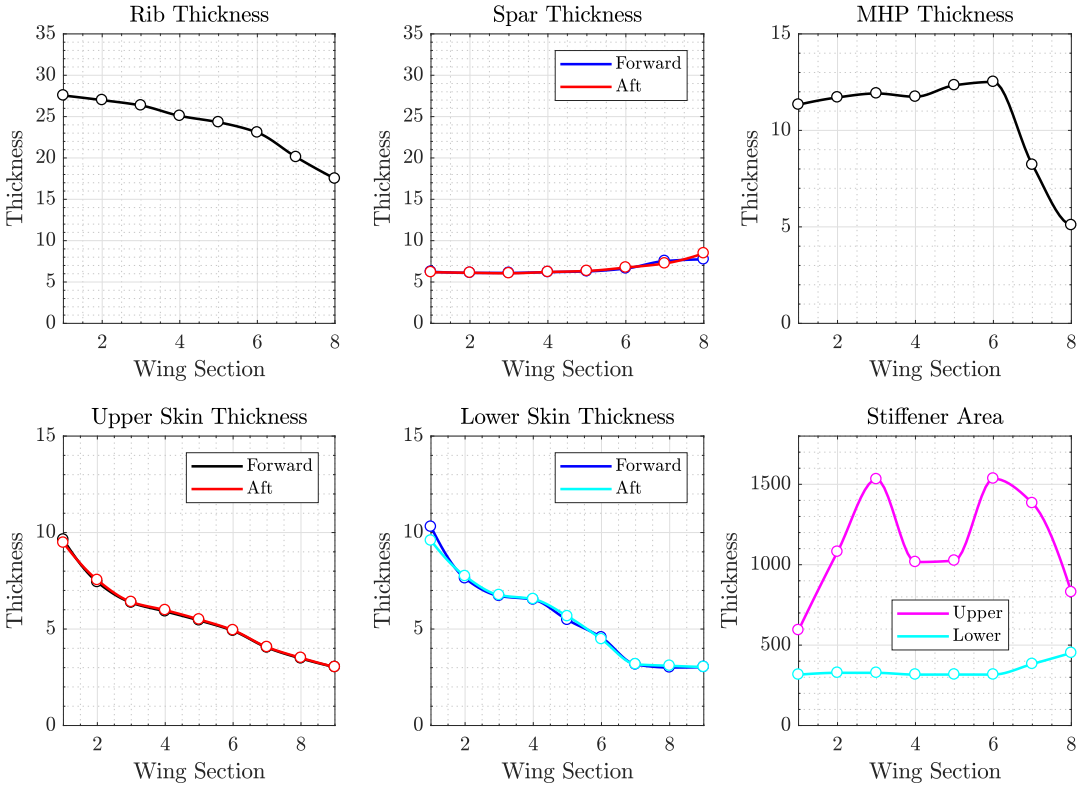
Figures 3.5(a) and 3.6(a) show the final design breakdowns for the best solutions to the optimistic and pessimistic design problems. Design parameters are presented per wing section, each representing approximately 2 m of the wingspan. In both the optimistic and pessimistic problems the system and linking variables feature a smooth variation root to tip, consistent with a conventional solution for a tip-loaded wing-box with a rectangular cross-section.

However, optimistic subsystem variables (stiffener dimensions and MHP sizing) feature some significant variations from the expected conventional solution. The upper panel stiffener area features peaks at both sections 3 and 7, with no relation to the variation in skin thickness. The MHP sizing also features a sharp decline from section 6 onward, with minimal variation elsewhere. The poor sizing of local variables is due to lower relative gradients for local subproblem responses; the sparse Jacobian formed for local responses using classical constraint decomposition; and a lack of active constraints driving local designs. The result is that although solutions satisfy all constraints they are far from the potential optimum design.

The sharp outboard variation in the MHP thickness has a smoother variation in the pessimistic design. Increasing the number of constraints on the system and enforcing more stringent local constraints leads to a more practical solution. Including a more comprehensive range of responses leads to a better overall design.

Figure 3.5(b) and 3.6(b) show the constraint distributions for optimistic and pessimistic design problems. Both designs meet all given constraints, with the pessimistic design featuring the upper and lower DFEM stress constraints. The common active constraints across the two problems are the spar buckling and upper panel buckling constraints. The pessimistic problem features lower DFEM stress constraints amongst the active constraints. The shift in active constraints to subproblems is consistent with the pessimistic assumptions.

(a) Design Variable Breakdown



(b) Constraint Value Distribution: 1441 Constraints

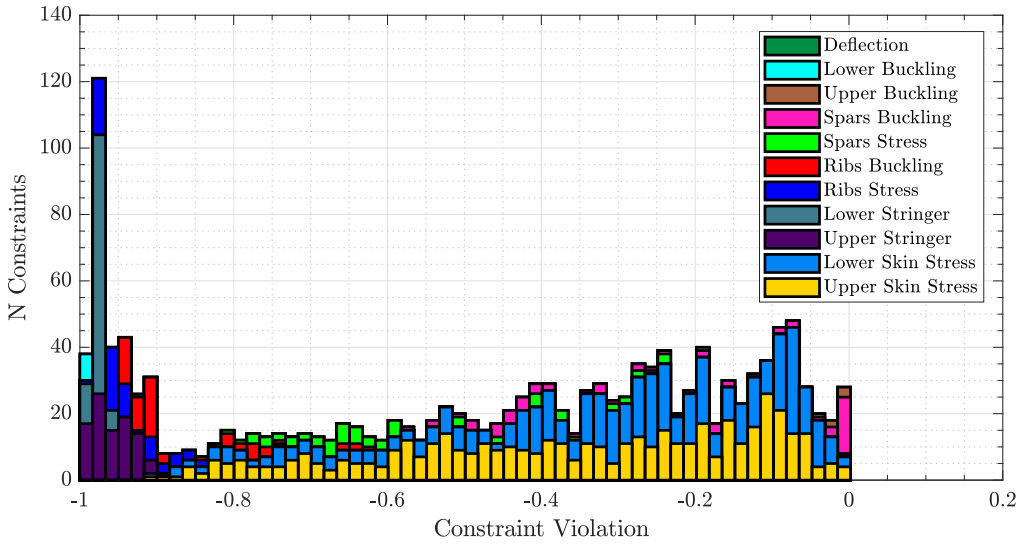
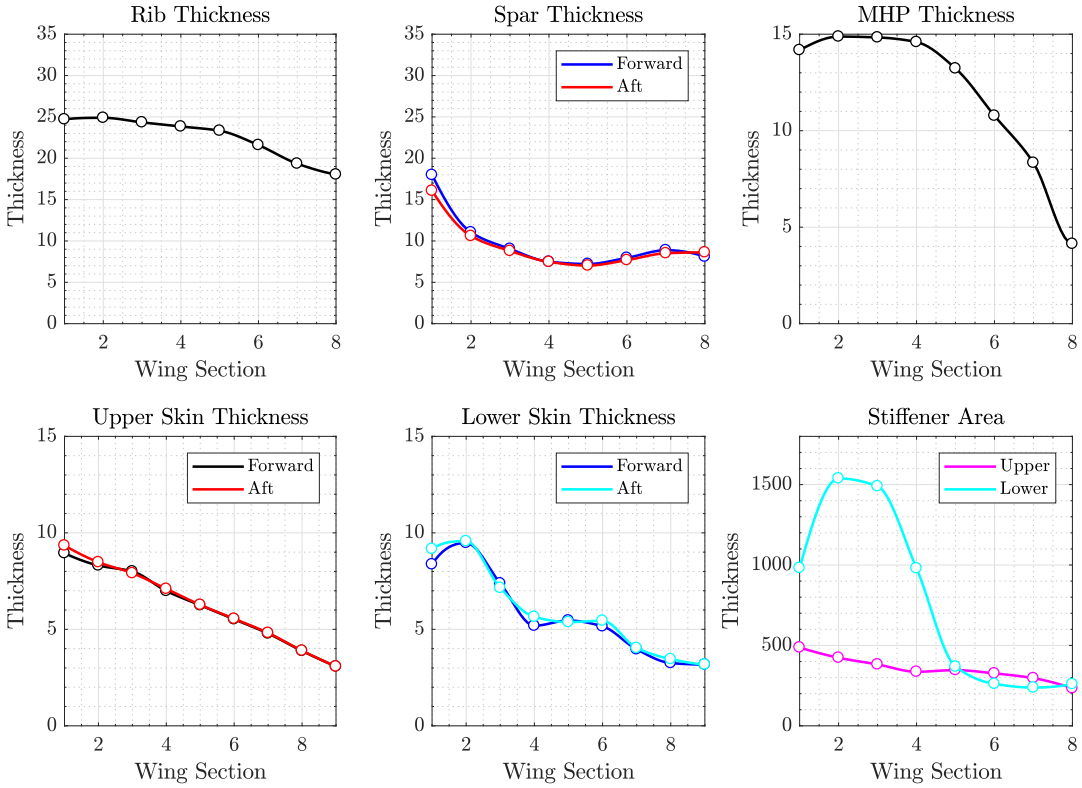


FIGURE 3.5. AAO Optimistic Problem Final Design Summary; Best selected design indicating the ribs, spar, upper skin, lower skin thickness' and upper-lower stiffener area per wing section. Final design, $M^* = 5717.2$ kg

(a) Design Variable Breakdown



(b) Constraint Value Distribution: 1945 Constraints

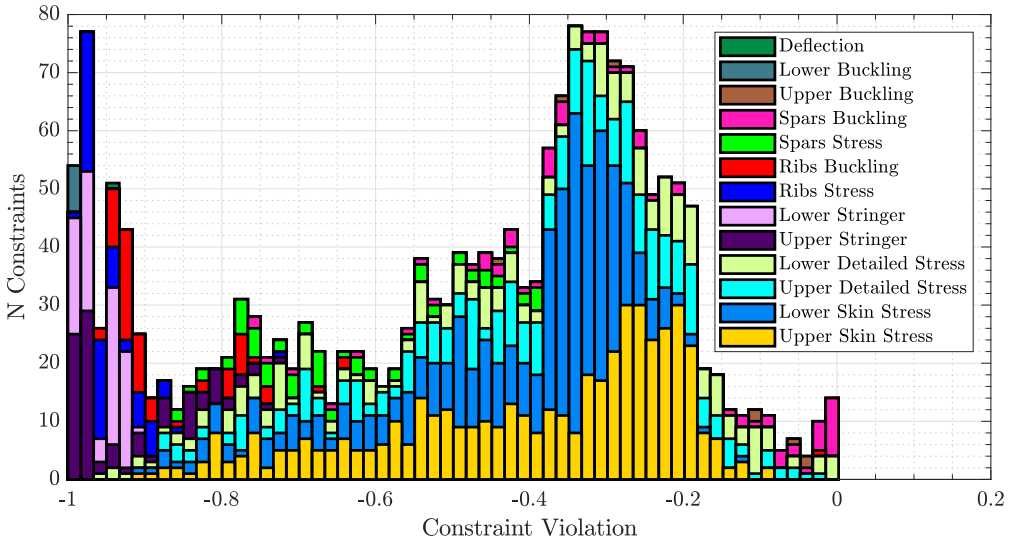


FIGURE 3.6. AAO: Pessimistic Problem Final Design Summary; Best selected final design. $M^* = 5925$ kg

(a) Optimistic Design Problem

(b) Pessimistic Design Problem

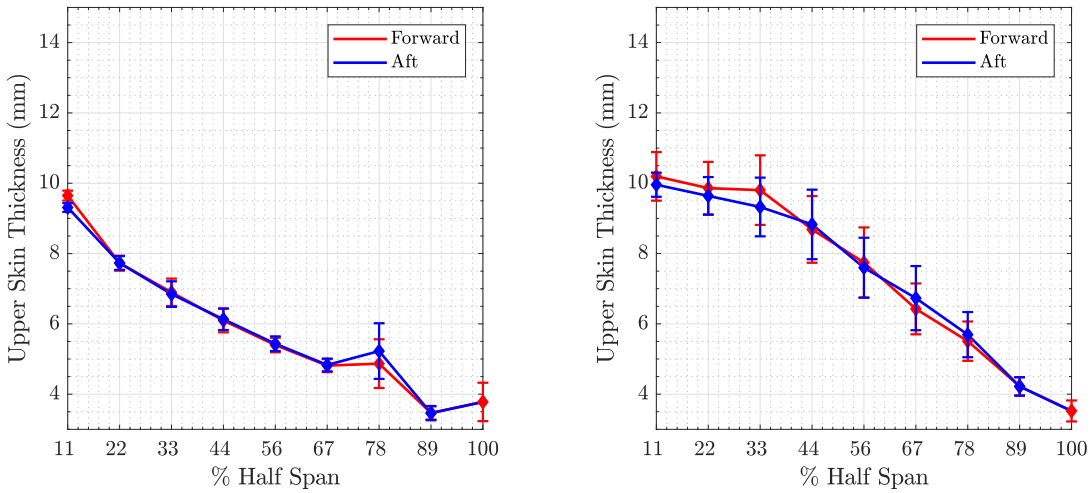


FIGURE 3.7. AAO: Upper Skin Design Variable Convergence; Distribution represents the mean solution from multiple start positions, with error bars representing variation in the converged design solutions.

Figure 3.7 shows how the upper skin cover design varies between the optimistic and pessimistic problems, with error bars indicating there relative variable variation between converged solutions from multiple starting positions. Additional local constraints in the pessimistic model have improved the subsystem design, but linking variables convergence on both the upper and lower skins have become less consistent. The pessimistic method features significant variation in the converged solutions across the majority of the wing. The optimistic solution only features variations in the converged solutions outboard along the span (with a higher number of tests completed), where model responses/constraints would be easier to satisfy than those at the root. This variation is indicative of how the pessimistic design assumptions affect the overall accuracy of the entire solution.

3.3 Multi-phase Optimisation Results

Figure 3.8 shows the convergence history for the SEQ design problem; a dashed line denotes each level. The level 1 problem converges after 28 iterations. For the level 1 analyses subsystem-level variables are at the lower bound, whereas each subsequent level uses the partial optima from the previous with multiple start positions for the newly added variables. This change compensates for the non-convex behaviour of additional constraints. At the boundary of each level, the additional subsystem variable variation causes a significant change in the objective function value and maximum active constraint.

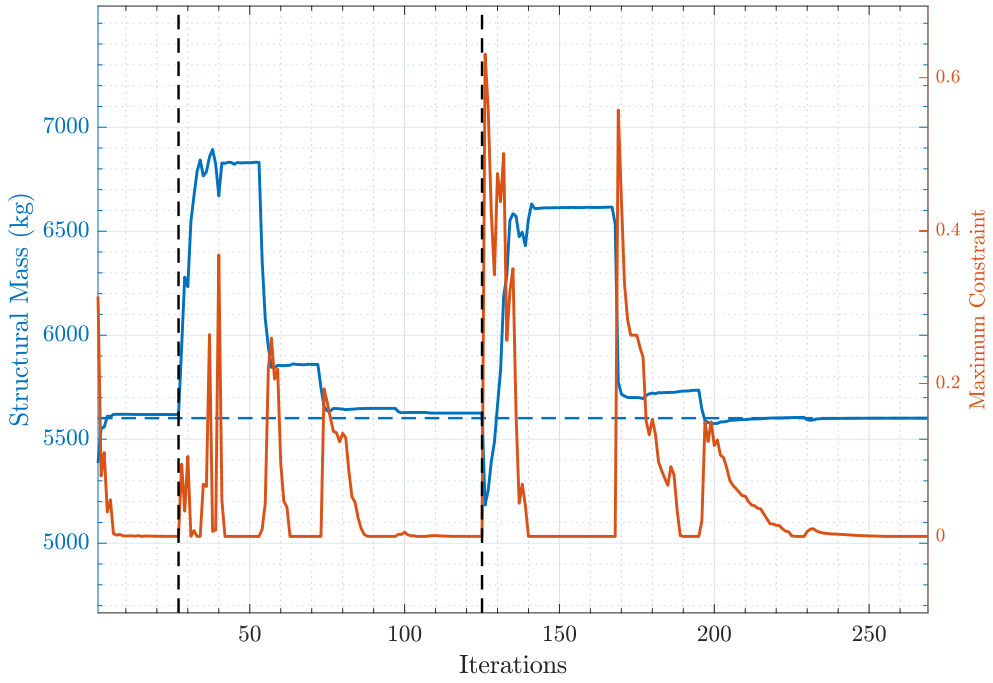


FIGURE 3.8. SEQ: Optimistic Problem Iteration History; Satisfactory less than zero maximum active constraint and final optimised objective mass = 5601.4 kg.

Figure 3.9 shows the objective iteration history for the SEQ strategy under optimistic and pessimistic assumptions. Figure 3.10 shows the MaxC iteration history for the same solutions. The shaded envelope between each level represents the difference between convergence histories with starting positions at the upper and lower bounds of the design space. The next level then starts from the best solution from the previous, in terms of objective and feasibility considerations.

(a) Optimistic Design Problem

(b) Pessimistic Design Problem

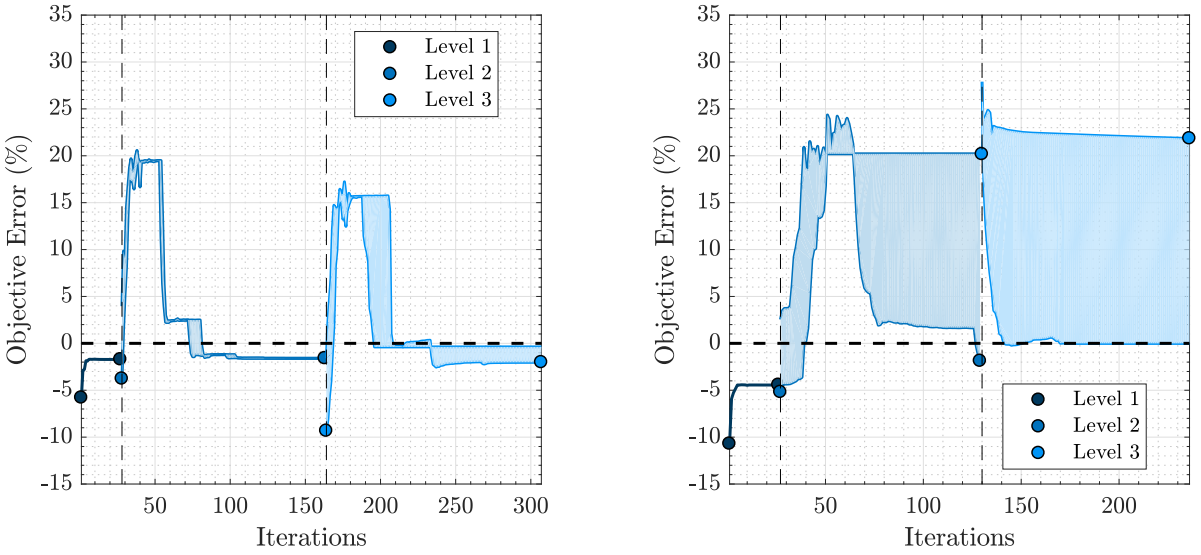
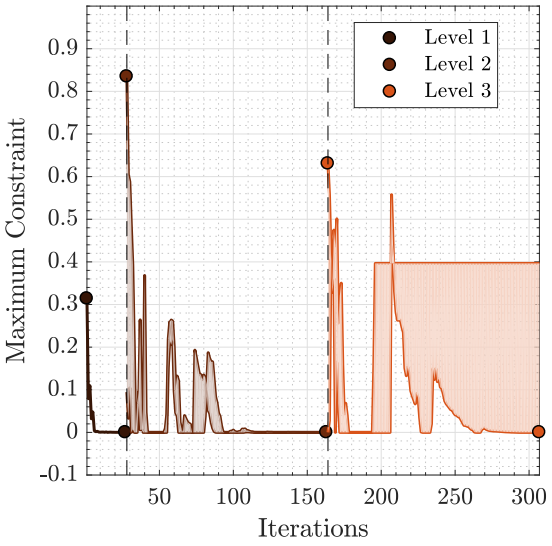


FIGURE 3.9. SEQ: Objective Function Iteration History; Objective is given in terms of normalised objective variation from M^*. Envelopes indicate maximum range of solutions from upper bound and lower bound starting positions for variables included in levels 2 and 3.

Both solutions exhibit non-convex properties with the addition of subsystems design problems. With each subsequent level, the difference between the converged solutions increases significantly. The larger the range of converged solutions at each level, the less likely that the design space is convex, and the less likely that a monolithic solver has found an accurate solution. The level 3 constraint envelope for the optimistic problem and the level 3 objective envelope in the pessimistic design problem both have a considerable variation in the convex solution space. The variation appearing in both the optimistic and pessimistic cases indicates that the problem of convexity is a more general problem within hierarchical multi-phase designs, regardless of the assumptions or difficulty of constraints.

The non-convex properties of the additional subproblems have a magnified effect on the convergence properties with each new level of detail added. In both the optimistic and pessimistic design space, the same problem occurs; when the given a starting position is a partial optimal solution to the majority of responses the optimiser is less likely iterate on the newer models added afterwards. Given a partial optimal solution, an equally optimal solution is not guaranteed with the addition of new levels of detail. The level 3 design problems have a significant envelope of solutions, when compared to the level 2 envelopes, indicating the problem is magnified when including models with more conservative margins of error.

(a) Optimistic Design Problem



(b) Pessimistic Design Problem

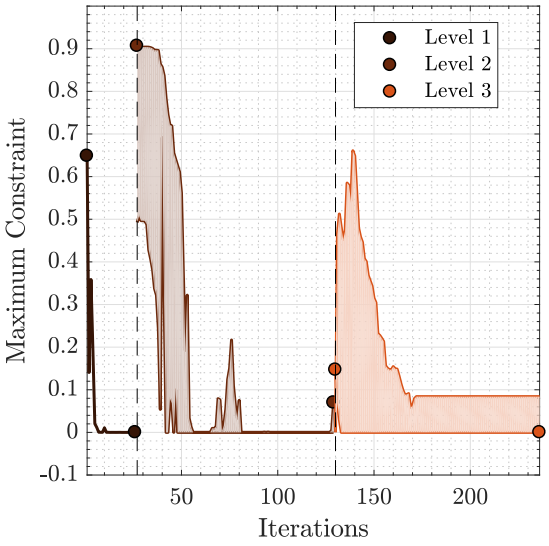


FIGURE 3.10. SEQ: MaxC Iteration History.

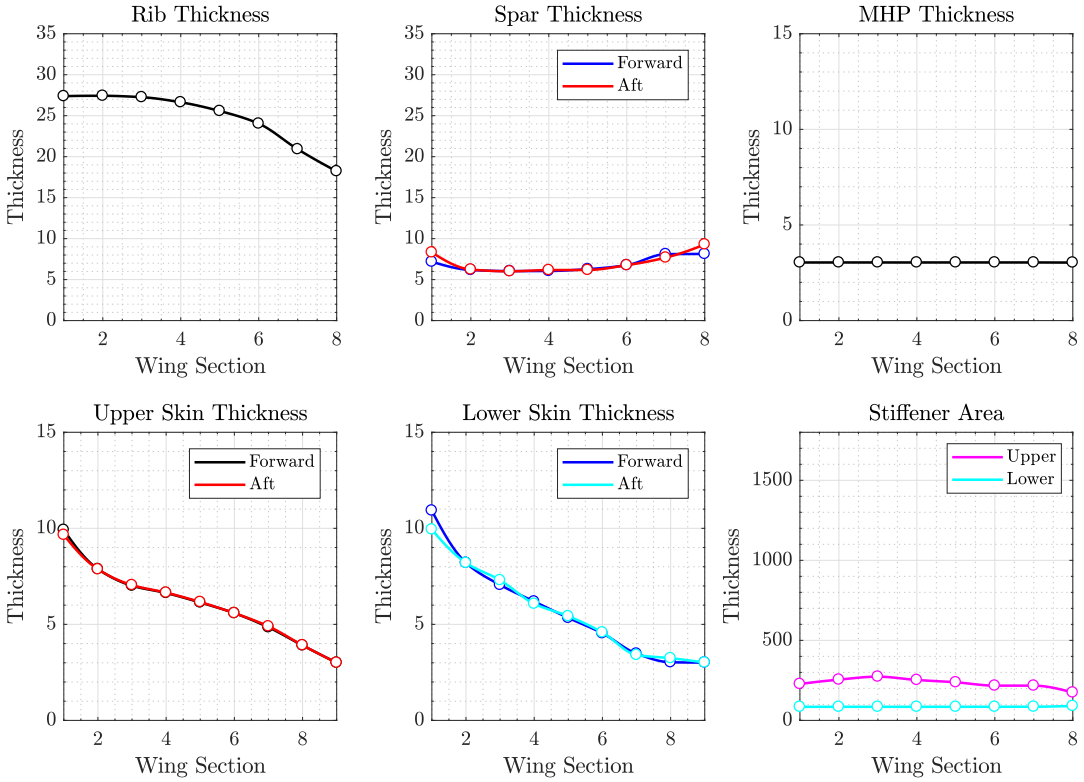
The SEQ method shows the extreme difference between the two final designs under different assumptions. Under optimistic assumptions, the final optimised objective mass shows a 5% improvement over the AAO solutions M^*. While under pessimistic assumptions the best final solution is $\approx$ 20% higher than the AAO solution, with a maximum active constraint of -0.1, far from an optimum, but still a non-changing converged solution.

Figures 3.11 and 3.12 are the final design summaries for the optimistic and pessimistic SEQ problems, respectively. The maximum active constraints for the pessimistic design problem indicate that, although the design is improved in comparison to the AAO method, there are still some over-designed sections on the wing.

Note the only multi-phase solver tested was sequential level mixing. There are a range of possible implementations of multi-phase solvers which may improve these results. There are many possible variants, three such methods may include:

- Gradual level mixing with static local gradients, updating procedurally through the design process, could improve this method but there is no mathematical basis or guarantee of improved converged proprieties.
- Gradient-based Level mixing with gradual local level optimisation subproblems gradually mixed in based on the variable and response sensitivities.
- Random level mixing schemes could also also reduce any potential biasing of methods to match the uniquely presented problem. Periodically adding or removing subproblems, retaining partial sub-optimal gradients for associated responses.

(a) Design Variable Breakdown



(b) Constraint Value Distribution: 1441 Constraints

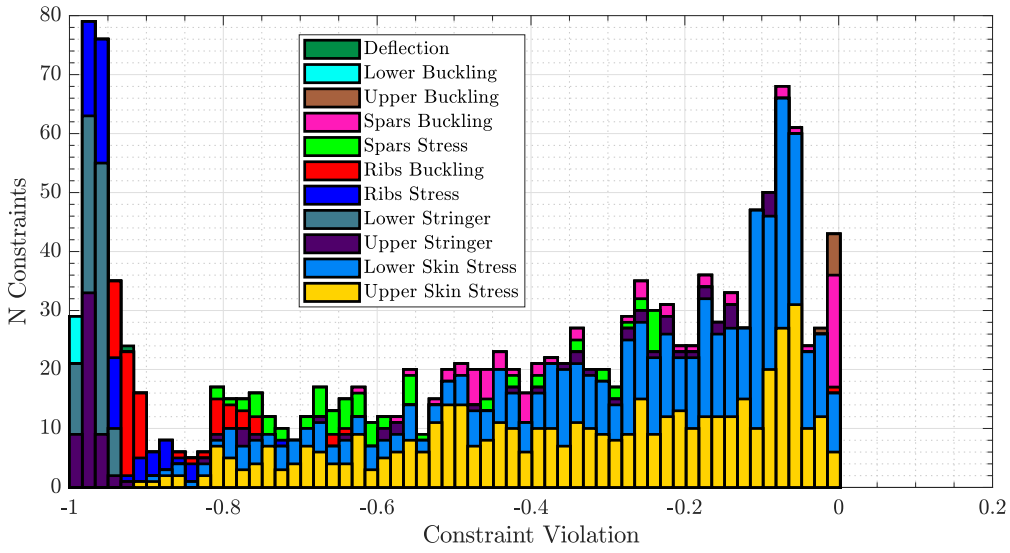
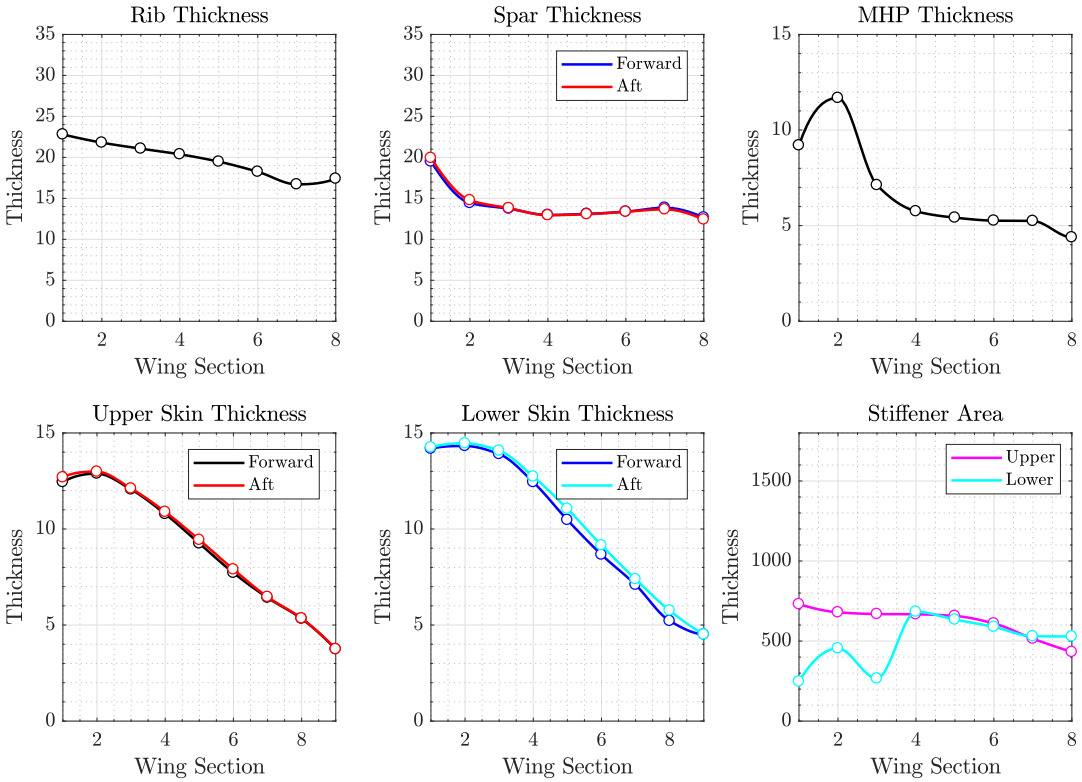


FIGURE 3.11. SEQ: Optimistic Problem Final Design Summary. $M^* = 5601.41$ kg

(a) Design Variable Breakdown



(b) Constraint Value Distribution: 1945 Constraints

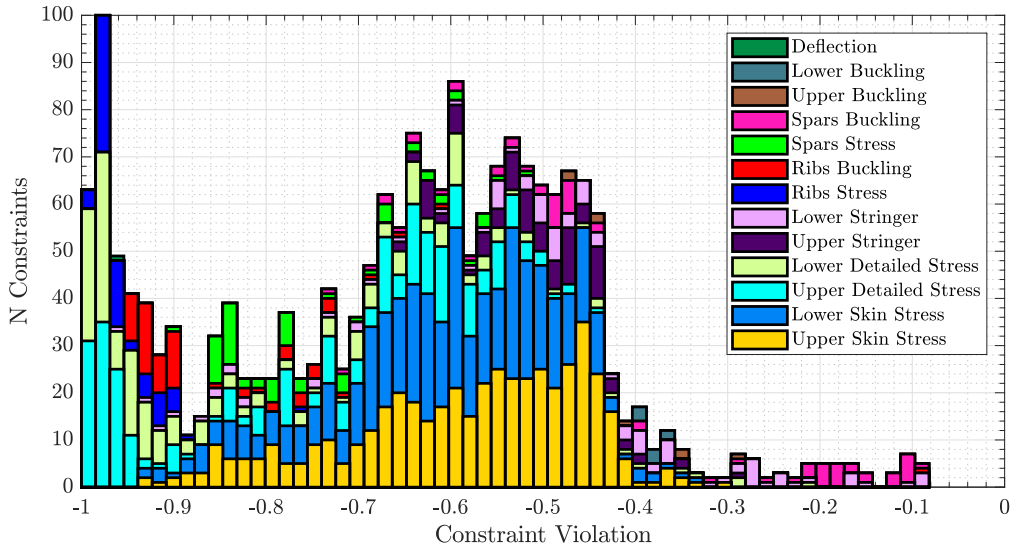


FIGURE 3.12. SEQ: Pessimistic Problem Final Design Summary. $M^* = 5716.77$ kg

3.4 Chapter Summary

This chapter has presented methods for single-phase and multi-phase monolithic optimisation procedures.

1. The single-phase method displayed a consistent solution under optimistic assumptions designs feature a maximum solution variation of 2.2%. Solutions found may satisfy constraints but are potentially far from the optimum design with local variables featuring deviations from the expected conventional solution.

2. Viable solutions were possible under pessimistic assumptions when using pessimistic assumptions, but repeated testing from multiple starting positions was necessary as 40% of solutions failed to converge. For the practical use in engineering design problems, the starting position using monolithic methods would need to be reasonably close to the expected design.

3. Additional design constraints at the local level reduced the reliability of the AAO monolithic method. However, including a more comprehensive range of responses leads to a better overall design.

4. The multi-phase solver converged to improved solutions over the single-phase AAO, but required significantly more function evaluations. The addition of subsystem problems introduced non-convex behaviour into solutions for both the optimistic and pessimistic design cases, magnifying convergence difficulties with subsequent levels.

5. Multi-phase solvers given a starting position is a partial optimal solution to the majority of responses, the optimiser is less likely iterate on the newer models added afterwards. Given a partial optimal solution, an equally optimal solution is not guaranteed with the addition of new levels of detail. This result has implications for the current design process of optimisation of local substructures after the preliminary design process has finalised the broader elements of the structural design.

The next chapter details the processes and results for a dual distributed optimisation architecture.

CHAPTER 4

DUAL DISTRIBUTED MULTILEVEL OPTIMISATION

This chapter presents the most up to date method within the dual distributed optimisation architecture: an augmented analytical target cascading method. This process uses an inner loop set of distributed optimisation subproblems, attempting to find a minimum with respect to local problem constraints, and an outer loop process coordinates the problems into a single consistent solution. The procedure is detailed with process descriptions and schematic. The results are presented under a range of design assumptions, with an additional state-parameter study.

Dual distributed optimisation procedures use local copies of linking variables between a set of isolated optimisation problems and are one of the significant subsets of MLO research. Testing the dual procedure will use one of the most widely used and developed methods with the literature, an augmented ATC with subproblems optimised in parallel.

4.1 Augmented Analytical Target Cascading

The augmented ATC uses a relaxed & alternating MLO strategy, allowing for the individual exploration of problems in-turns, and slowly ramping consistency penalties towards a converged solution. Broadly the ATC algorithm uses the following procedure [110, 138, 144]:

Step 1: Partition the given design variables and penalty parameters.
Step 2: *Inner Loop* optimisation, solving for the decomposed problem.
Step 3: If the inner loop constraints and consistency constraints meet some tolerance, the optimisation has converged. If so the solution has converged, otherwise step 4.
Step 4: The *Outer Loop* checks if consistency constraints are not met or system: subsystem solutions are infeasible coupling and state-parameters must be updated.
Step 5: Update starting positions using the most recent solutions, return to Step 1.

In step 2, the inner loop solves a set of optimisation problem, the generic form of dual problems are in equation (4.1) and (4.2). Here, $i = 1, \cdots, N$ represents the local subproblem identifier. *Consistency constraints*, q, enforce conformity between the system and subsystems, here presented as some equality constraint function of the system-level linking variables, $\boldsymbol{x}_0$, and their equivalent duplicates, $\hat{\boldsymbol{x}}_0$.

$$
\begin{aligned}
\underset{x,x_0}{\text{minimise}} \quad & F_0(\boldsymbol{x},\boldsymbol{x_0}) = M(\boldsymbol{x},\boldsymbol{x_0}) \\
\text{subject to} \quad & \sigma_{\text{rf},1-m}(\boldsymbol{x},\boldsymbol{x_0}) \leq 0, \\
& \sigma_{\text{rf},1-p}(\boldsymbol{x},\boldsymbol{x_0}) \leq 0, \\
& \delta_{\text{rf}}(\boldsymbol{x},\boldsymbol{x_0}) \leq 0, \\
& \phi_0.c^c(\boldsymbol{x},\boldsymbol{x_0}) \leq 0, \\
& q_0(\boldsymbol{x_0},\hat{\boldsymbol{x_0}}) = 0 \\
& x^{\text{L}}, x_0^{\text{L}} < \boldsymbol{x},\boldsymbol{x_0} < x^{\text{U}}
\end{aligned}
\tag{4.1}
$$

$$\begin{aligned} \underset{x_0,y}{\text{minimise}} \quad & f_i(\hat{\boldsymbol{x}}_{\boldsymbol{0}}, \boldsymbol{y_i}) = M_i(\hat{\boldsymbol{x}}_{\boldsymbol{0}}, \boldsymbol{y_i}) \\ & \lambda_{\mathrm{rf},i}(\hat{\boldsymbol{x}}_{\boldsymbol{0}}, \boldsymbol{y_i}) \leq 0, \\ & \sigma_{\mathrm{rf},istr}(\hat{\boldsymbol{x}}_{\boldsymbol{0}}, \boldsymbol{y_i}) \leq 0, \\ & \sigma_{\mathrm{rf},iMHP}(\hat{\boldsymbol{x}}_{\boldsymbol{0}}, \boldsymbol{y_i}) \leq 0, \\ & q_i(\boldsymbol{x_0}, \hat{\boldsymbol{x_0}}) = 0 \\ & x_0^{\mathrm{L}}, y_i^{\mathrm{L}}, < \hat{\boldsymbol{x}}_{\boldsymbol{0}}, \boldsymbol{y_i} < x_0^{\mathrm{U}}, y_i^{\mathrm{U}} \end{aligned} \tag{4.2}$$

In step 4, the outer loop, some weighted factor on penalty function, q, is adjusted to either enforce or relax consistency following some update strategy. Initially, the need for consistency amongst all variables is relaxed, allowing for unhindered individual model exploration. As the algorithm progresses, the weighting factor on consistency penalties increases, tightening the coupling between subproblems.

Consistency is increasingly enforced throughout until a feasible and coherent design is found. Allowing individual problem exploration while incrementally increasing consistency penalties ensures a top-down objective and bottom-up constraint solution sequence. Top-down objectives of the system-level problem are transferred to subproblems by resolving for minimal consistency penalties; while bottom-up constraints transfer the need for subsystem constraint feasibility through consistency penalties on the system-level problem.

4.1.1 ATC with Augmented Lagrangian Coordination

The outer-loop update strategy is critical to the efficiency of ATC. The most efficient form of strategy within the literature applies consistency using an augmented Lagrangian-based quadratic penalty function. All duplicated linking variables synchronise with the system-level solution, which in itself contains all linking variables, ensuring all subproblems converge on a single coherent solution. Each linking variable has an associated Lagrangian multiplier and a weighted consistency factor. The Lagrangian updates are relative to an alternating direction method of multipliers used to relax/tighten Lagrange terms to properly scale subproblem penalties. The weighted consistency factor is relative to a simple ramp function, increasing linearly.

Variable Copies and Consistency Penalties: Each subproblem contains duplicates of one or more linking variables. The ultimate goal of the consistency constraints is to ensure these are equal by the end of the optimisation process. The final solution should be coherent. Practically this means linking variables, $\boldsymbol{x_0}$, must be consistent in both system and subsystem problems, matching the primal form of the design problem.

The equality constraint introduced uses the squared difference between the real and the local variable copy; this is given in equation (4.3). As long as the consistency constraints, $\boldsymbol{q}$, are satisfied or are within some tolerance, the solution can be considered feasible for the primal problem [138].

$$\boldsymbol{q_i}(\boldsymbol{x_0}, \tilde{\boldsymbol{x_0}}) = |\tilde{\boldsymbol{x_{i0}}} - \boldsymbol{S}_i \cdot \boldsymbol{x}_0|^2 = 0 \tag{4.3}$$

Here, $\boldsymbol{q_i}$ represents the local variable discrepancy. *Linking variable scarcity* means that none of the subproblems depends on all of the linking variables; if this were the case, it would be a system-level response. $\boldsymbol{S}_i$ is a binary selection matrix used to allocate the correct linking variables for subsystem i. Initial selection matrices are based on physical panels, similar to how the division of component design is undertaken in industry. The linking variables selected were the immediate upper and lower skin thickness' with an effect on the local panel sizing. Likewise, $\tilde{\boldsymbol{x_{i0}}}$ are the local copies of those linking variables for subsystem i.

Lagrangian Penalty Relaxation and Coupling: ATC uses a relaxed penalty-based approach allowing for optimisation subproblems to be entirely separable. Once separated, subproblems can be solved concurrently and independently of one another. ATC enforces consistency constraints through a series of local penalty functions. Augmenting the penalty functions with weighted factors and Lagrangian multipliers allows scaling of coupled problems and gradually ramping penalties. The resulting augmented penalty function is given in equation (4.4):

$$\phi_i(\boldsymbol{\mu_i}, \boldsymbol{q_i}) = \boldsymbol{\mu_i}^T \boldsymbol{q_i} + \frac{1}{2}||\boldsymbol{w_i} \circ \boldsymbol{q_i}||^2 \tag{4.4}$$

Where $\phi_i(\boldsymbol{\mu_i}, \boldsymbol{q_i})$ represents the consistency penalty function, $\boldsymbol{\mu_i}$ represents the vector of Lagrange multipliers and $\boldsymbol{q_i}$, represents the local inconsistency for the ith subproblem. Local weighting factor, $\boldsymbol{w}_i$, is a state-parameter set outside the optimisation problem. Note the symbol $\circ$ represents the Hadamard product: the term by term multiplication of vectors, such that $\boldsymbol{a} \circ \mathbf{b} = [a_1, ...a_n]^T \circ [b_1, ...b_n]^T = [a_1 b_1, ..., a_n, b_n]^T$ [145].

The generic augmented form of the objective function, f_L, is based on well-established theorems of Lagrangian duality problems. Where, f_L, is equivalent to the original subproblem objective plus an equality consistency constraint and a weighted error term, resulting in equation (4.5):

$$\underset{v,\mu}{\text{minimise}}\ f_L(\boldsymbol{v}, \boldsymbol{\mu_i}) \quad = \quad f_i(\boldsymbol{v}) + \phi_i(\boldsymbol{\mu_i}, \boldsymbol{q_i}) \quad = \quad f_i(\boldsymbol{v}) + \boldsymbol{\mu_i}^T q(\boldsymbol{v}) + \frac{1}{2}|\boldsymbol{w_i} \circ \boldsymbol{q}(\boldsymbol{v})|^2 \tag{4.5}$$

Lagrangian multipliers, $\boldsymbol{\mu}$, are applied to deviant solutions as a "penalty", for some positive consistency $\boldsymbol{q}$, weighted with $\boldsymbol{w}$. Within duality theory, the augmented Lagrangian is equivalent to the standard problem (4.6):

$$\begin{aligned} \underset{v,\mu}{\text{minimise}} \quad & f_L(\boldsymbol{v},\boldsymbol{\mu}) = f_i(\boldsymbol{v}) + \frac{1}{2}|\boldsymbol{w} \circ \boldsymbol{q}(\boldsymbol{v})|^2 \\ \text{subject to} \quad & \boldsymbol{q}(\boldsymbol{v}) = 0 \end{aligned} \tag{4.6}$$

The an additional term $\frac{1}{2}|\boldsymbol{w} \circ q(\boldsymbol{v})|^2$ does not change the objective optimal solution point for the optimal Lagrangian multiplier, μ^*. As inconsistency and weighting tends to nil ($q(\boldsymbol{v}) \Rightarrow 0$, $(\boldsymbol{w}) \Rightarrow 0$) the problem returns to the original subproblem statement without coupling, or penalties [107, 146].

4.1.2 Inner Loop Decomposed Problem Solution

The final decomposed solution takes the form of equation (4.7) for the system-level and equation (4.8) for the subsystem-level problem. These combine the original problem statement for the objective functions and includes the new penalty quadratic penalty function, ϕ.

$$\begin{aligned} \underset{x,x_0}{\text{minimise}} \quad & F_0(\boldsymbol{x},\tilde{\boldsymbol{x_0}}) + \sum_{i=1}^{N} \phi_i(\boldsymbol{\mu_i}, \boldsymbol{q_i}(\boldsymbol{x},\tilde{\boldsymbol{x_{i0}}})) \\ \text{subject to} \quad & c_0(\boldsymbol{x},\boldsymbol{x_0}) \leq 0 \end{aligned} \tag{4.7}$$

$$\begin{aligned} \underset{\tilde{x_0},y_i}{\text{minimise}} \quad & f_i(\boldsymbol{y_i},\tilde{\boldsymbol{x_0}}) + \phi_i(\boldsymbol{\mu_i}, \boldsymbol{q_i}(\boldsymbol{x_0},\tilde{\boldsymbol{x_{i0}}})) \quad \text{for } i = 1,...N \\ \text{subject to} \quad & c_i(\boldsymbol{y_i},\tilde{\boldsymbol{x_0}}) \leq 0 \end{aligned} \tag{4.8}$$

4.1.3 Outer Loop Update Strategy

The dual problem is solved through appropriate selection of values for the Lagrange multiplier, μ, aiming to reduce the relaxation error in an out-of-the-loop iterative process. The correct estimate of μ^* is one which balances solving for the primal consistency and set of feasible locally optimal subproblems. This iterative approach assumes a feasible optimum exists in the design space, meeting both consistency and feasibility tolerances.

Stopping Conditions: A general criterion for outer loop convergence is when the system and subsystem level problems represent a feasible and coherent solution. A convergence criterion can be set as equation (4.9), where the variable, $||\boldsymbol{q}||_\infty$, is the norm of variable inconsistencies in the between all subproblem linking variables and there associated copies.

$$||\boldsymbol{q}^k - \boldsymbol{q}^{k-1}||_\infty < \epsilon \text{ and } ||\boldsymbol{q}^k||_\infty < \epsilon \tag{4.9}$$

Here k is for the k-th solution step, and ϵ represents a very small convergence tolerance.

Penalty Parameter Update: Setting the initial weighting factors for $\boldsymbol{\mu}^{(0)}$ and $\boldsymbol{w}^{(0)}$ are non-trivial. As before, incorrectly balancing the need for local exploration with primal consistency constraints can lead to significant inefficiencies. These can occur because the algorithm allows for too much individual exploration before convergence; or because limiting local problems to consistent solutions too soon misses potentially viable solutions in the design space.

The generalised method updates Lagrangian multipliers to scale penalties, $\boldsymbol{\mu_i}$, using an alternating direction method of multipliers. Scaling occur in the outer loop using the current consistency constraints to set the multipliers for next step using equation (4.10) [138, 147].

$$\boldsymbol{\mu}^{k+1} = \boldsymbol{\mu}^{k} + 2\boldsymbol{w}^{k} \circ \boldsymbol{w}^{k} \circ \boldsymbol{q}^{k} \tag{4.10}$$

Here, $\boldsymbol{\mu}^{k+1}$ is the complete vector of Langragian multipliers to be used in the next phase, based on the current weightings, $\boldsymbol{w}$, and concatenated consistency constraints, $\boldsymbol{q}$. The initial value is set to $\mu^{(0)} = [0]$ to allow for the first step to find locally optimal solutions before iterating towards the coupled problems optima.

The initial weighting will be set based on the methods used in non-linear programming. With initial weighting based on equation (4.11).

$$w_i^{(0)} = \sqrt{\frac{\alpha \hat{f}_i}{\hat{\boldsymbol{q}}_{\boldsymbol{i}}{}^{T} \hat{\boldsymbol{q}}_{\boldsymbol{i}}}} \tag{4.11}$$

Where $\alpha \hat{f}_i$ is a sum fractional estimate of the typical objective function, $\alpha = 0.1$. $\hat{\boldsymbol{q}}_{\boldsymbol{i}}$ is obtained by the initial consistency value for an optimum system: subsystem solution without a penalty function, i.e. $\hat{\boldsymbol{q}}_{\boldsymbol{i}} = \boldsymbol{q}\{F_0(x^*, x_0^*)$ and $f_i(x_{i0}^*, y_i^*)\}$ for each subsystem $i = 1, \ldots N$.

The direct penalty weighting is updated through a simple ramp function, increasing the base penalty based on current consistency constraint violations, equation (4.12).

$$\boldsymbol{w_i}^{k} + 1 = \begin{cases} \boldsymbol{w_i}^{k} & \text{if} \quad |\boldsymbol{q_i}|^{k} \leq \gamma |\boldsymbol{q_i}|^{k-1} \\ \beta \boldsymbol{w_i}^{k} & \text{if} \quad |\boldsymbol{q_i}|^{k} > \gamma |\boldsymbol{q_i}|^{k-1} \end{cases} \quad \text{for } i = 1, \ldots N \tag{4.12}$$

Here β is the ramp rate for the weighted consistency multipliers, some value greater than 1. While γ is the tolerance check that the consistency has improved with the current iteration, some value less than one. This study will use a nominal value of $\beta = 4$ and $\gamma = 0.4$, based on initial testing.

4.2 Augmented ATC Process Description

The final process descriptions is shown in algorithm 3. The steps outlined on the right-hand side broadly correspond with the initial design process outlined at the start of this chapter. An intermediate step was added to include the delineation of system: subsystem-level optimisation for the inner loop problem solvers. Figure 4.1 shows the schematic for the ATC procedure, with the corresponding process steps labelled.

Algorithm 3 ATC Optimisation Procedure

Inputs Starting position: $[v^{(0)}]$, Bounds: v^{L}, v^{U}
Set initial weights, $\boldsymbol{w}^{(0)}$ and $\boldsymbol{\mu}^{(0)}$; ▷ Step 1
repeat $3 \Rightarrow 23$
    Convert v to model sizing x, x_0 and, local panel sizing $y_{1,\cdots,N}$;
    Create Equivalent laminate skin properties;
    parfor $i = 1 \Rightarrow N$ **do** ▷ Step 2a
        Make local copies of variables $\hat{x_{i0}}^{(0)} = S_i x_0^{(0)}$
        function FMINCON($[\hat{x_{i0}}^{(0)}, y_i^{(0)}]$, $[x_0^{\mathrm{L}}, y_i^{\mathrm{L}}]$, $[x_0^{\mathrm{U}}, y_i^{\mathrm{U}}]$);
            Subsystem model: run DFEM `SOL 200`, $\models f_i(\hat{x_{i0}}, y_i)$ and $c_i(\hat{x_{i0}}, y_i)$;
            Penalty Functions Evaluations $\phi_i(q_i, \mu_i, w_i)$;
        return Partial solution: $[\hat{x_{i0}}^*, y_i^*, q_i^*]$;
    end parfor
    Create Equivalent laminate skin properties;
    function FMINCON($[x^{(0)}, x_0^{(0)}]$, $[x^{\mathrm{L}}, x_0^{\mathrm{L}}]$, $[x^{\mathrm{U}}, x_0^{\mathrm{U}}]$); ▷ Step 2b
        System model: run GFEM `SOL 200` $\models F(x, x_0)$ and $c_0(x, x_0)$;
        Penalty Functions Evaluations $\phi_0(q_0, \mu_0, w_0)$;
    return Partial solution: $[x^*, x_0^*, q_0^*]$
    Concatenate Distributed Variables: $\hat{x_0}^*, c_{0\cdots N}^*, q_{0\cdots N}^*$;
    Test Stopping Conditions: System-Subsystem Converged Feasible ▷ Step 3
    if Stopping Conditions $\neq$ true **then**
        Update Penalty weightings and Lagrange Multipliers $\boldsymbol{w}^{k+1}, \boldsymbol{\mu}^{k+1}$ ▷ Step 4
        Update Starting Positions: $v^{(0)} = [x*, \hat{x_0}^*, y_{1\cdots N}^*]$ ▷ Step 5
    else System: Subsystem Design Conformity: $[\hat{x_0}^*] = [x_0^*]$
until $3 \Rightarrow 23$: Stopping Conditions = true
return Converged solution: $v^* = [x^*, x_0^* y_{1,\cdots,N}^*]$

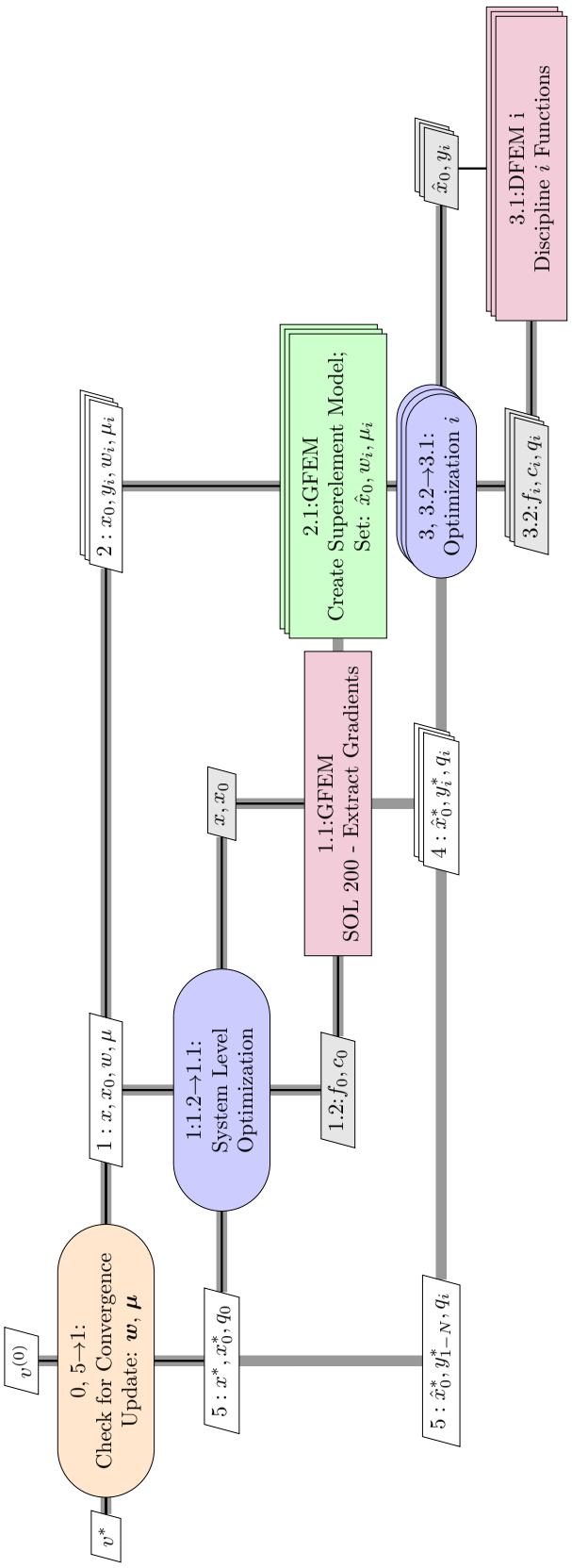


FIGURE 4.1. Diagram for the ATC procedure

4.3 Dual Distributed Optimisation Results

The augmented ATC method shown here is a relaxed alternating optimisation architecture. The inner loop optimisation problem solves for the constrained design problem, while the outer loop coordinates and scales the overall problem. Figure 4.2 shows a summary of the outer loop information for the optimistic ATC procedure. Each outer loop iteration represents a series of independent system and local subproblem converged solutions.

The constraints and model consistency all converge to a feasible and coherent solution after 7 outer loops, with a multiplier ramp rate, $\beta = 4$. The final objective mass was 5521.5 kg, 3.5% lighter than the AAO solution. The procedure required a total of 18658 function evaluations compared to 5950 using the AAO method, over 3 times higher.

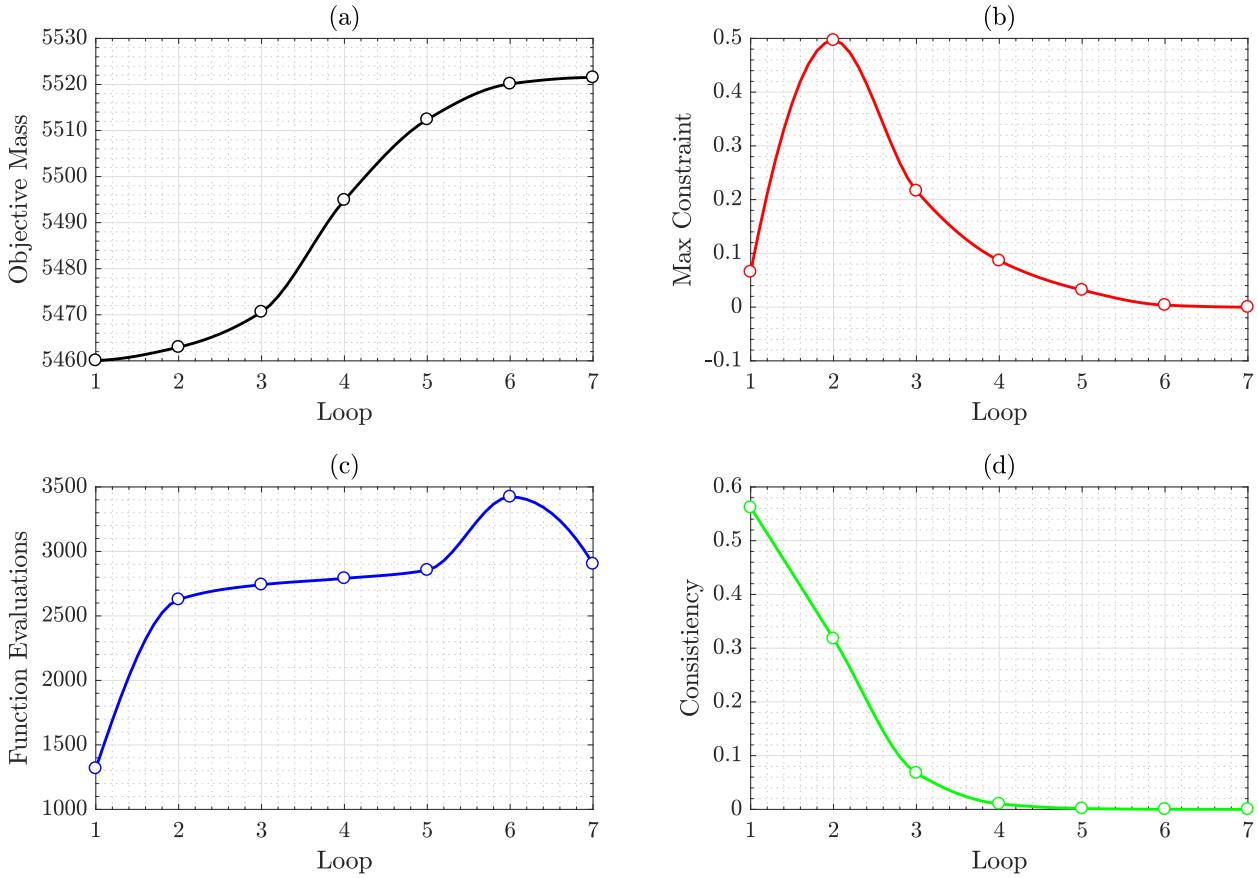


FIGURE 4.2. ATC: Optimistic Outer Loop Optimisation Results Summary
(a) Objective Mass, (b) Maximum Active Constraint,
(c) Function Evaluations per Loop, (d) Model Consistency

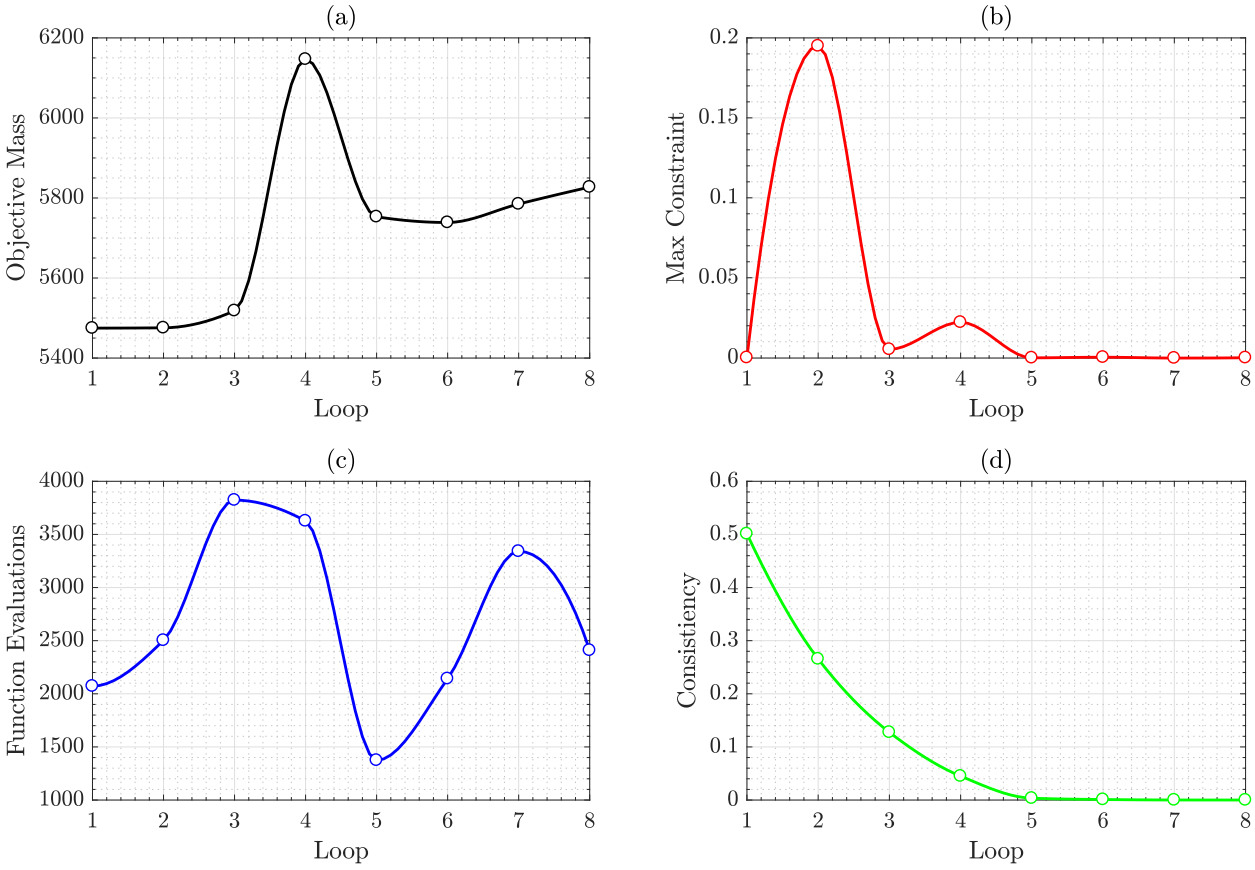


FIGURE 4.3. ATC: Pessimistic Outer Loop Optimisation Results Summary;
(a) Objective Mass, (b) Maximum Active Constraint,
(c) Function Evaluations per Loop, (d) Model Consistency

Figure 4.3 is the summary for the pessimistic outer loop problem with ramp rate, $\beta = 4$. Note the objective mass and number of function evaluations per loop do not show a cumulative improvement in optimisation performance. Each loop represents a different set of solution spaces, where the previous solution is the starting point for the next and subproblems are then solved independently.

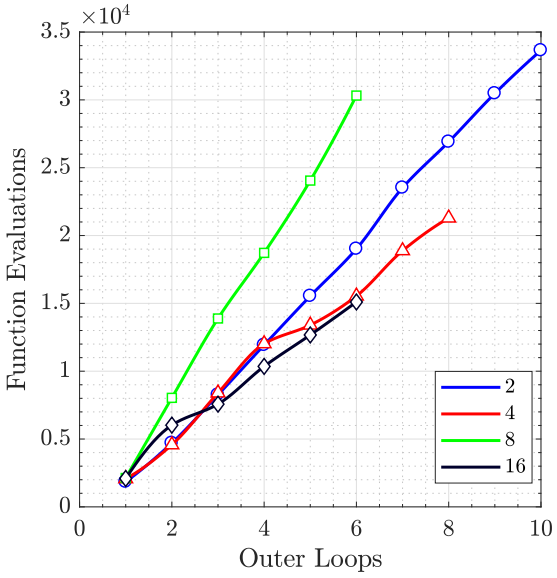
The final objective mass for the ATC pessimistic problem was 5827.2 kg a 4% increase against the best presented AAO pessimistic solution. Similar to the optimistic process, the ATC procedure required significantly more function evaluations than the monolithic methods, 21291 function evaluations, ~ 3.5 times higher than that of a single AAO based solution.

4.3.1 ATC State-parameter Study

For the presented problem, the ATC procedure can generate a consistent solution under both optimistic and pessimistic assumptions. However, ATC represents a significant increase in computation, with overlapping optimisers linked by a series of state-parameters. Parameters which have a significant effect on the overall procedural behaviour of the system, such as decomposition selection matrices, S_i, ramp rate, β, starting weight and starting Lagrangian multipliers, $w^{(0)}, \mu^{(0)}$. State-parameters will be varied and tuned to show their overall impact on the design process.

Consistency Penalty Ramp Rate: Table 4.1 shows how the pessimistic problem changes with ramp rate β, with the total function evaluations and objective variation history shown in figure 4.4. Overall there is non-trivial relationship between an increase ramp rate and the total number of function evaluations. If the ramp rate is set too low, the solution takes an increased number of outer loops, but it may not converge in the correct part of the design space. However, when the ramp rate is too high, the solution can be curtailed into a consistent solution before reaching a globally optimal solution.

(a) Function Evaluations v Outer Loops



(b) Objective Variation v Function Evaluations

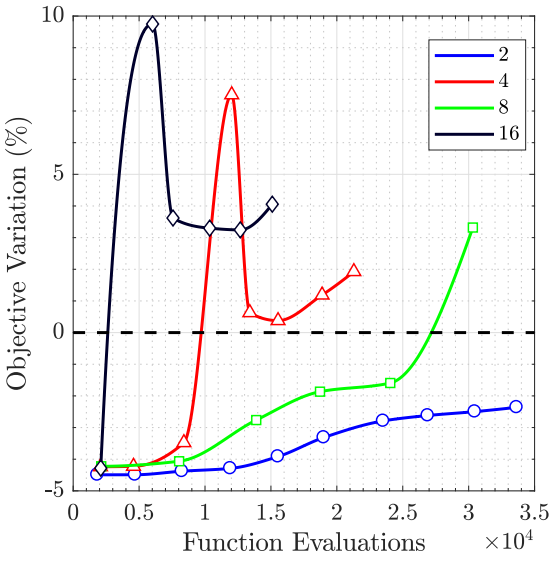


FIGURE 4.4. ATC: Pessimistic Problem Efficiency Variation with Ramp Rate: β

TABLE 4.1. ATC: Pessimistic Problem Solution Variation with Ramp Rate (β)

Ramp Rate, β	2	4	8	16
Objective Mass (kg)	5582.109	5827.218	5906.82	5948.98
MaxC†	0.201	0.000	0.000	0.006
Outer Loops	10	8	6	6
Function Evaluations	33647	21291	30318	15103

Gradient-based Partition: The partition selection matrices, S_i, were based on physical panels. The resulting being only using variables with a direct effect on the sizing/ mass. A gradient-based method would not differentiate between system and linking variables and, instead, select those which meet some sensitivity threshold, similar to the variable dependence measures used in the sub-space approximation studies [79]. Non-negligible gradients were dynamically selected in the outer loop process, modifying the selection matrices to include any variable with gradient responses of a similar or greater order of magnitude than the mean, shown in equation (4.13). In practise this means within some order of magnitude tolerance, ϵ, dictated by the scale of the mean gradient.

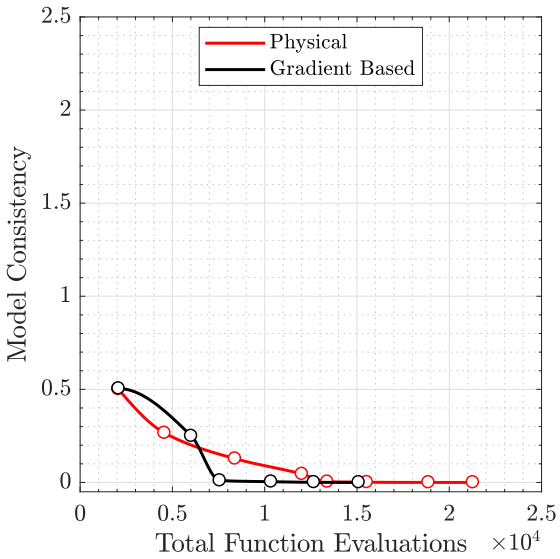
$$\boldsymbol{S_i} = \begin{cases} \left(|\frac{dc_i}{dx}| > \sim mean(|\frac{dc_i}{dx_0}|)\right) \\ \left(|\frac{df_i}{dx}| > \sim mean(|\frac{dc_i}{dx_0}|)\right) \end{cases} \quad \text{for } i = 1,...N \tag{4.13}$$

Table 4.2 shows how the design problem changes with selection matrices S_i, between the different partitioning methods. The gradient-based method led to expanded (Ex.) variable selection matrices, taking into account a greater number of linking variables for each local problem. Total function evaluations against model consistency histories are shown in figure 4.5. The gradient-based partition approach leads to significantly fewer function evaluations, approximately 30% and 15% fewer function evaluations using optimistic and pessimistic assumptions, respectively. In both cases, the final designs were only marginally different using the gradient-based method.

TABLE 4.2. ATC: Pessimistic Physical v Gradient-based Partition Strategy ($\beta = 4$)

Ramp Rate, β = 4	Optim.	Ex. Optim.	Pess.	Ex. Pess.
Objective Mass (kg)	5521.521	5494.762	5827.218	5948.982
MaxC	0.000	0.000	0.000	0.006
Outer Loops	7	6	8	6
Function Evaluations	18658	15878	21291	15103

(a) Optimistic Problem



(b) Pessimistic Problem

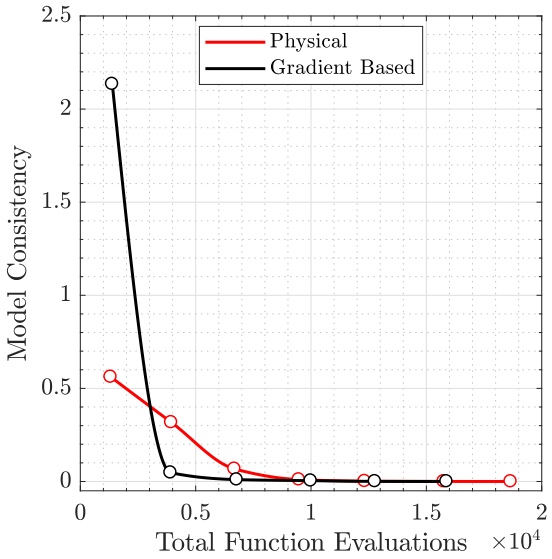


FIGURE 4.5. ATC: Physical v Gradient-based Partition Strategy ($\beta = 4$)

Local Optimisation Algorithm: ATC is a strictly feasible MLO architecture, meaning that for every outer loop iteration the local subproblems must be locally feasible. Based on this requirement, the local subproblems in the current tests use a multi-start optimisation, which in turn leads to a significant increase in function evaluations. In order to test how vital the local optimisation solution is the subproblem algorithm was replaced with a single-start optimiser (a regular `fmincon` SQP-based algorithm) for comparison. Figure 4.6 and table 4.3 show the overall performance.

The MaxC is consistently unsatisfactory in the single start design case, possibly due to the solution being weighted towards unfavourable parts of the design space while attempting to maintain system: subsystem consistency. The resulting performance shows that using the multi-start optimisation strategy doubles the function evaluations for the same number of outer loops when compared to the single-start method. The significantly fewer design iterations bring the overall performance to a comparable expense to the AAO solutions presented. Indicating that, given a strictly convex local optimisation problem, the two procedures may not have as much of a difference. However, if the local optimisation strategy were strictly convex, a similar AAO solution may also be comparatively simpler to solve.

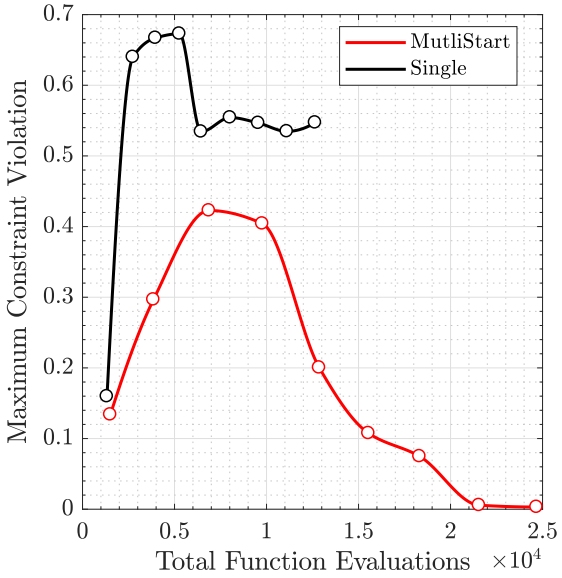


FIGURE 4.6. ATC: Optimistic Single start v Multi-start Local optimisation ($\beta = 2$)

Ramp Rate, β = 2	Single	Multi-start
Objective Mass (kg)	5472.1023	5478.464
MaxC	0.547	0.003
Outer Loops	9	9
Function Evaluations	12645	24680

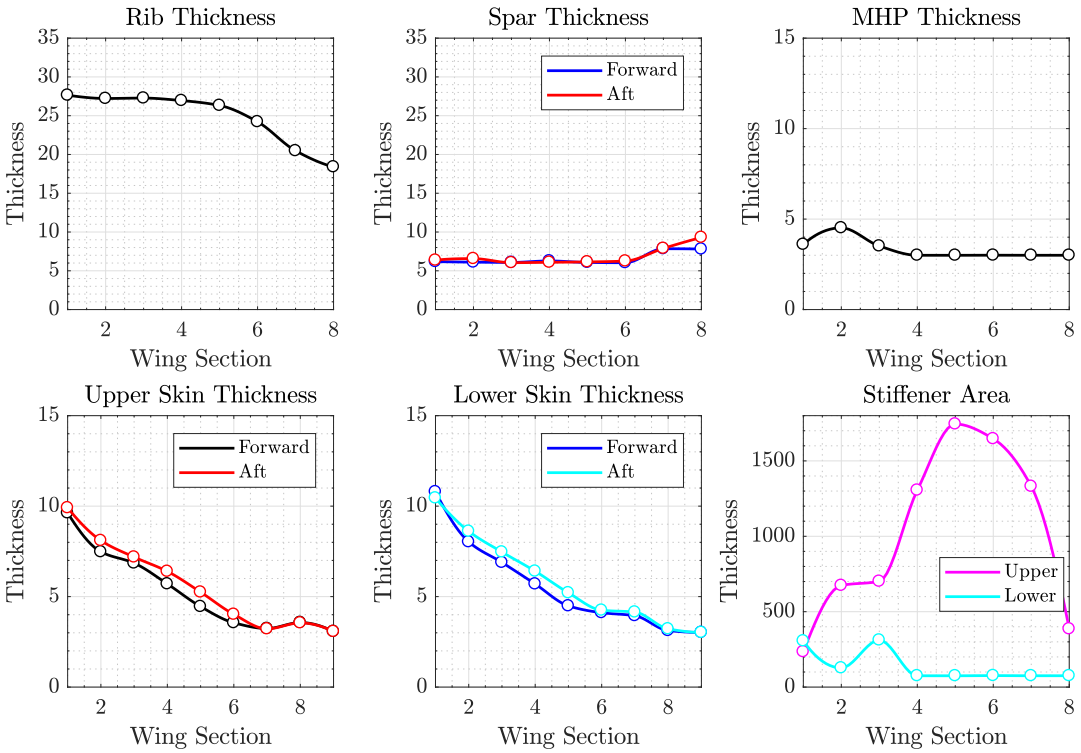
TABLE 4.3. ATC: Optimistic Single v Multi-start Local Strategy ($\beta = 2$)

4.3.2 Augmented ATC Design Summary

Figures 4.7 and 4.8 show the best calculated design summaries for the optimistic and pessimistic solutions using the augmented ATC procedure. In the optimistic procedure without the additional DFEM constraints, the local MHP and stiffener areas are poorly optimised, but the overall design is a significant improvement when compared to the AAO solution to the same problem. The pessimistic solutions shows less differences between the final AAO and ATC design, but ATC still represents an improved design with improved local tailoring on the upper and lower surfaces.

The final solutions have a pronounced forward: aft skin thickness variation. This differential was due to the control point locations for stress sampling. Stress control points were the same for all algorithms presented, but the ATC solution is the only algorithm which discovered the discrepancy. The difference between the AAO and ATC local solutions is due to the local model being optimised in isolation, with the resulting defects in modelling methods becoming more apparent in the final design.

(a) Design Variable Breakdown



(b) Constraint Value Distribution: 1441 Constraints

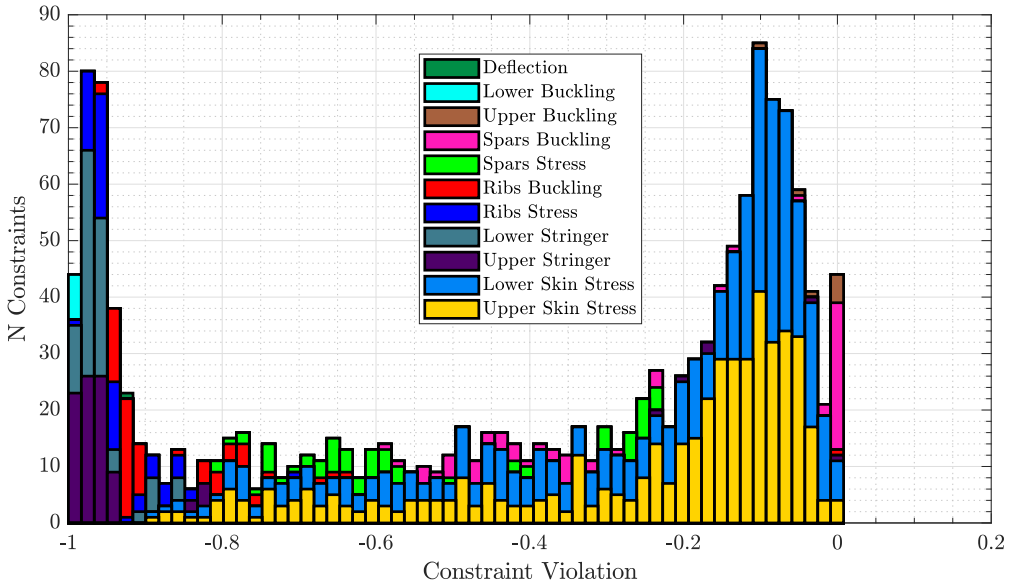
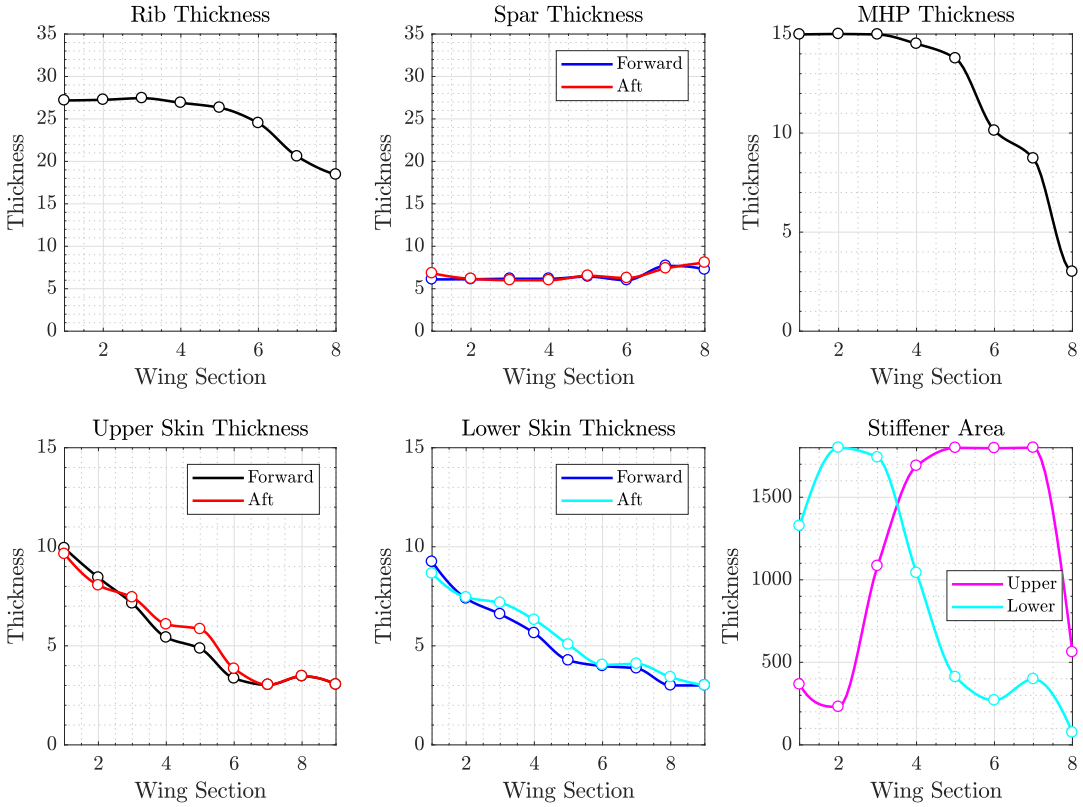


FIGURE 4.7. ATC: Optimistic Problem Final Design Summary; Gradient-based Partition, $\beta = 2$, $M^* = 5478.46$ kg

(a) Design Variable Breakdown



(b) Constraint Value Distribution: 1945 Constraints

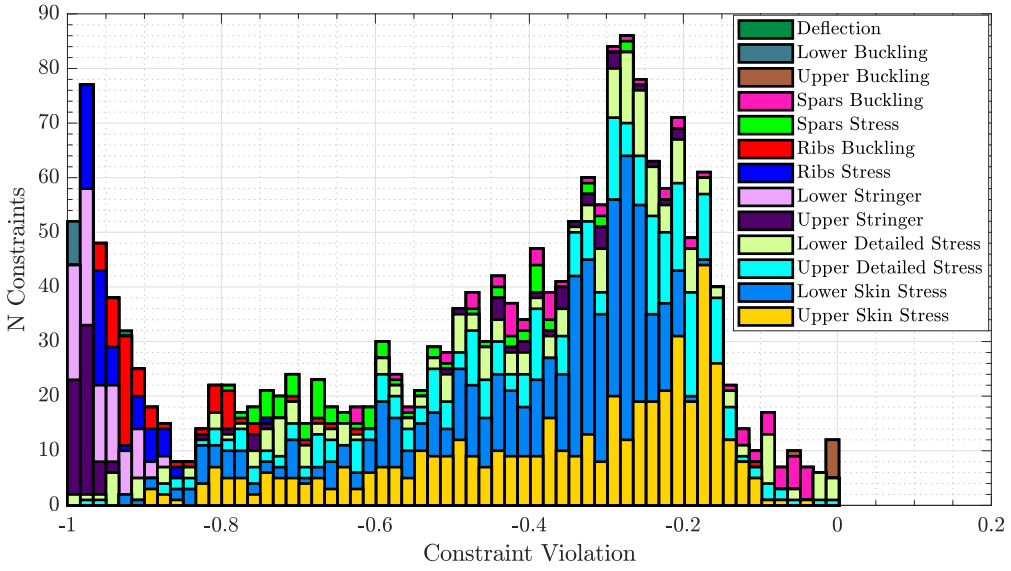


FIGURE 4.8. ATC: Pessimistic Problem Final Design Summary; Gradient-based Partition, $\beta = 4$, $M^* = 5827.21$ kg

4.4 Chapter Summary

This chapter has outlined the dual distributed method for ATC with augmented Lagrangian coordination. This method was tested on a set of design problems, with a state-parameter study into outer loop ramp rate and types of partition.

Overall it was found that:

1. The dual distributed augmented ATC can produce designs significantly better than monolithic solvers, but at a cost of significantly more function evaluations.

2. The ATC procedure accuracy is heavily dependent on the governing state-parameters, it is often unclear how to set these values, requiring trial & error to properly tune them.

3. A gradient-based partition improved the overall solution by ignoring physical definitions of linking variables. MLO within the same design system and discipline needs to be more reactive to the system-level design as often the system: subproblem design relationships are non-trivial.

4. Increasing the number of local constraints in the pessimistic design case improved the overall design and altered the necessary values of state-parameters, favouring a more aggressive outer loop design process.

The next chapter presents primal distributed optimisation, the last subset of multilevel optimisers presented within the literature review.

CHAPTER 5

PRIMAL DISTRIBUTED MULTILEVEL OPTIMISATION

Primal distributed architectures represent the last significant subset of multilevel optimisation results considered in this thesis. These optimisation architectures strictly maintain the primal form of the distributed problem, with no duplication of linking variables or responses. In order to compensate for reduced local information in subproblems, primal methods use more robust coordination strategies for system: subsystem coupling. This chapter outlines two illustrative methods using primal distribution with both nested and alternating formulations considered.

Primal MLO algorithms, as discussed in chapter 1.4.2 on page 29, do not duplicate linking variables present in distributed subproblems. The result being subproblems have a limited range of design variables to affect the local responses, in response, primal methods have been applied to more straightforward numerical problems, and have much more substantial coordination strategies for the system-level problem.

The primal optimisation procedures tested in this section will be based on the most mature methods presented within the literature for a nested and an alternating formulation. The nested formulation will use a *surrogate BLISS* procedure; using a surrogate-based interaction approximation method, instead of the original sub-gradient method. The alternating formulation will use an *ADMM* procedure; based on the virtual Stackelberg approach and the simplest primal optimisation procedure to implement with a robust coordination method.

A coupled post-optimality approach was considered but could not be tested. These algorithms would have been prohibitively computationally expensive within the scope of this research. Coupled methods require the accurate calculation of the Jacobian within a nested architecture, a process which would have been overly cumbersome to include, given the scope of this investigation.

Note, on computational limitations and prohibitive costs there were technical limitations in transferring the models and functions used here into Linux based high-performance computer. Further studies may wish to start from a base of methods readily transferable to high-performance machines or, in-time, repeat this and other studies in the context of constant growth in computing power.

5.1 Surrogate BLISS Procedure

This section will detail an iterative MLO problem structure based on the BLISS architecture. The original form of BLISS was updated from primal to a dual architecture in the form of BLISS-2000; the new architecture was more efficient and capable of solving more difficult problems. The procedure used here was meant to test primal architecture, specifically those adapted to use surrogate models or local artificial functions.

Surrogates generated in the concept design process could be stored and used as functions to maintain design freedom for longer in the design process, however this only works if surrogates can reliably replicate local behaviour. The use of surrogates also improves the robustness of interaction approximation models for non-convex subproblems, compared to the original sub-gradient method. Using sub-gradients to approximate local behaviour works best with a reliably convex solution space where the magnitude and sign of gradients are somewhat consistent. Using locally produced and reflexively updated surrogates can both aid later design whilst also better approximating non-convex solution spaces.

The surrogate enhanced BLISS method is as follows:

Step 1: Solving a distributed and concurrent set of subsystem local optimisation problems.

Step 2: Generate a surrogate-based interaction approximation to couple subsystem optimal design selection with linking variables.

Step 3: A surrogate coupled system-level optimisation, using an interaction model to approximate subproblem solutions.

Step 4: If surrogate error at the optimum point meets some tolerance, the optimisation has converged. Otherwise, update existing surrogates with additional points to improve accuracy and return to step 2.

5.1.1 Interaction Approximation Model

The first step in developing a surrogate based interaction model is to define the models which are interacting. In this case, the interaction is between the system-level model and the subsystem-level model. The MLO problem uses quasi-separable assumptions, meaning that the only elements of interaction are the linking variables, defined as $\boldsymbol{x}_0$. The subsytem level models are defined using both these linking variables and a set of local design variables, $y_{1,\cdots,N}$, for subproblems up to N subproblems.

The resulting interaction between system: subsystem level models can then be described using only linking variables as the sole means of input. The interaction model then approximates the optimum local subsystem design variables based on said input. This interaction model model will use the surrogate model in-order to nest this process into the system-level optimisation.

Subsystem Optimisation: Step 1 of the process is solving a distributed set of local optimisation problems. The local optimisation results will inform the starting position for the system-level coordination problem. Using the EGO method defined in chapter 2.5.3 on page 55, this initial local exploration stage will output the baseline surrogates used in the next stage of the algorithm.

By using the EGO in the first instance to generate the baseline surrogates allows:

(a) A direct comparison of the total computational expense of all methods, including the effort taken to build surrogates. Previous research has not included the surrogate generation, which in some instances may take substantial time and cost.

(b) Outputting a suitable surrogate of the local subproblem is a synergistic side-effect for the next step in the MLO algorithm. Effectively using EGO to solve the local problem concatenates the expense for local optimisation and surrogate model development - ideally improving efficiency.

The EGO will solve a subsystem-level optimisation with respect to local and linking variables as seen in equation (5.1), where M_i is the local panel mass taken from the DFEM.

$$\begin{aligned} \underset{x_{i0},y_i}{\text{minimise}} \quad & f_i(\boldsymbol{x_{i0}},\boldsymbol{y_i}) = M_i(\boldsymbol{x_{i0}},\boldsymbol{y_i}) \quad \text{where section: } i = 1,\cdots,N \\ & \bar{c}_i(\boldsymbol{x_{i0}},\boldsymbol{y_i}) \leq 0 \\ & x_0^L, y^L < \boldsymbol{x_{i0}},\boldsymbol{y_i} < x_0^U, y^U \end{aligned} \tag{5.1}$$

Interaction Approximation: Step 2 of the algorithm is to reformulate the surrogates generated during the EGO method into an interaction model. The interaction model approximates the optimum local design variables, $\boldsymbol{y}^*{}_i$, for a given input of liking variables, $\boldsymbol{x}_{i0}$.

Local surrogates are numerical functions, where the number of evaluations are inconsequential in comparison to the rest of the algorithm. Using this fact the local interaction model will use a hybrid GA, optimising the local model response with respect to local variables, with fixed linking variables. The resulting optimum local variables, $\boldsymbol{y}_i^* = \hat{\boldsymbol{y}}_i(\boldsymbol{x}_{i0})$, parse through the real subsystem model. The interaction model then outputs the real function responses given with respect to linking variables and optimised local variables, as shown in equation (5.2).

$$\left.\begin{aligned} M_i(\boldsymbol{x}_{i0},\hat{\boldsymbol{y}}_i(\boldsymbol{x}_{i0})) &\models M_i^*(\boldsymbol{x}_{i0},\boldsymbol{y}_i^*) \\ \bar{c}_i(\boldsymbol{x}_{i0},\hat{\boldsymbol{y}}_i(\boldsymbol{x}_{i0})) &\models \bar{c}_i^*(\boldsymbol{x}_{i0},\boldsymbol{y}_i^*) \end{aligned}\right\} \quad \text{for } i = 1,\cdots,N \tag{5.2}$$

The final step of the interaction model is to bring the local subsystem responses into the system-level problem. The real function responses, M_i and $\bar{c}_i$, and associated gradients from `SOL 200` are outputs from the interaction model. These gradients include the post-optimality sensitivity gradients on local responses with respect to linking variables, $\frac{d\bar{c}_i}{dx_{i0}}$.

The artificial functions used will be consistent and updated throughout to reduce surrogate errors with each design step. Generating intermediate surrogate models for local optima with each iteration will be too computationally expensive to maintain primal decomposition as a competitive alternative to monolithic approaches. While pre-allocated, it would be disingenuous to directly compare the efficiency of each architecture without including the computational expense of developing surrogates.

Coupled System Level Optimisation: The final phase is to solve the fully coupled system-level optimisation. Each system-level design step is associated with a set of coordinated subproblems based on the previously established artificial interaction approximation models. As a whole, the algorithm maintains the primal form of the problem, with no duplicates of linking variables or responses in the final algorithm. The coupled system-level optimisation statement given in equation (5.3).

$$
\begin{aligned}
\underset{x,x_0}{\text{minimise}} \quad & F_0(\boldsymbol{x},\boldsymbol{x_0}) = M(\boldsymbol{x},\boldsymbol{x_0}) + \sum_{i=1}^{N} \tilde{M}_i(\boldsymbol{x_0},\hat{\boldsymbol{y}}_{\boldsymbol{i}}(\boldsymbol{x_{i0}})) \\
\text{subject to} \quad & \sigma_{\text{rf},1-m}(\boldsymbol{x},\boldsymbol{x_0}) \leq 0, \\
& \sigma_{\text{rf},1-p}(\boldsymbol{x},\boldsymbol{x_0}) \leq 0, \\
& \delta_{\text{rf}}(\boldsymbol{x},\boldsymbol{x_0}) \leq 0, \\
& \bar{c}_i(\boldsymbol{x},\boldsymbol{x_{i0}},\hat{\boldsymbol{y}}_{\boldsymbol{i}}(\boldsymbol{x_{i0}})) \leq 0, \quad \text{for } i = 1,...N \\
& x^{\text{L}} < \boldsymbol{x},\boldsymbol{x_0} < x^{\text{U}}
\end{aligned}
\tag{5.3}
$$

Stopping Conditions: The final design must meet two conditions: the combined design variable vector $v = [x, x_0, y_{1,\cdots,N}]$ must have converged to a consistent solution; and the surrogate functions used at the current design point must be accurate.

The output of the coupled formulation results in a local variables still in the artificial form $y_i = \hat{y}_i(x_{i0})$. Therefore if the design is at the true global optimum, the artificial function must be reasonably close to the real function value. The global optimum should be when the RSME at the current design point is with some given tolerance, and the final solution has converged to a consistent solution where the design can no longer be improved. The resulting stopping conditions is equation (5.4), with given tolerances defined as ϵ.

$$
\begin{aligned}
||\boldsymbol{v}^k - \boldsymbol{v}^{k-1}||_\infty < \epsilon \\
\text{RSME}(\boldsymbol{x}*,\boldsymbol{x_0},\hat{\boldsymbol{y}}_{\boldsymbol{i}}(\boldsymbol{x_0}) < \epsilon
\end{aligned}
\tag{5.4}
$$

Until the algorithm meets the given stopping conditions the surrogate models are reflexively updated with information generated from the past phase. The design points, $\hat{y}_i$, plus an additional set of design points found using some surrogate improvement in-fill criteria will be tested to retrain existing surrogate models [139]. The coupled system-level optimisation will start again using the previous optima as the new starting position.

The coupled system-level optimisation and surrogate retraining will then continue iterating between one another until the stopping conditions are met.

5.1.2 Surrogate BLISS Process Description

The XDSM for the surrogate-based BLISS architecture can be seen in figure 5.1. The process description is given in algorithm 4.

Algorithm 4 BLISS Optimisation Procedure

Inputs Starting position: $[v^{(0)}]$, Bounds: v^{L}, v^{U}
parfor $i = 1 \Rightarrow N$ **do**
    Define local variables $x_{i0}^{(0)} = S_i x_0^{(0)}$
    function EGO($[x_{i0}^{(0)}, y_i^{(0)}]$, $[x_0^{\mathrm{L}}, y_i^{\mathrm{L}}]$, $[x_0^{\mathrm{U}}, y_i^{\mathrm{U}}]$);
        Subsystem model: run DFEM `SOL 200`, $\models f_i(x_{0i}, y_i)$ and $c_i(x_{0i}, y_i)$; ▷ Step 1
        Surrogate Constructor: $\tilde{f}_i(x_{0i}, y_i)$ and $\tilde{c}_i(x_{0i}, y_i)$;
    return Surrogate: $\tilde{f}_i, \tilde{c}_i$ and Partial solution: $[x_{0i}^*, y_i^*, q_i^*]$;
end parfor
repeat $13 \Rightarrow 26$
    Concatenate Distributed Variables: $x_0^*, y_{1\cdots N}^*$;
    Update Starting Positions: $x_0^{(0)} = x_0^*$
    Rearrange Surrogates to resemble artificial functions with inputs: $x_{i0}, \hat{y}_i(x_{i0})$ ▷ Step 2
    Convert v to model sizing x, x_0 and, local panel sizing $y_{1,\cdots,N}$;
    Create Equivalent laminate skin properties;
    function FMINCON($[x^{(0)}, x_0^{(0)}]$, $[x^{\mathrm{L}}, x_0^{\mathrm{L}}]$, $[x^{\mathrm{U}}, x_0^{\mathrm{U}}]$); ▷ Step 3
        System model: run GFEM `SOL 200` $\models F(x, x_0)$ and $c_0(x, x_0)$;
        for $i = 1 \Rightarrow N$ **do**
            Surrogate Functions Evaluations $\tilde{f}_i(x_{i0}, \hat{y}_i(x_{i0})), \tilde{c}_i(x_{i0}, \hat{y}_i(x_{i0}))$;
        System model: run GFEM `SOL 200` $\models F(x, x_0)$ and $c_0(x, x_0)$;
        parfor $i = 1 \rightarrow N$ **do**
            Extract subsystem solutions: $y_i = \hat{y}_i(x_{i0}))$
            Subsystem model: run DFEM `SOL 200` $\models f_i(x_{i0}, y_i)$ and $\bar{c}_i(x_{i0}, y_i)$
        end parfor
    return Solution: $[x^*, x_0^*, \hat{y_{1\cdots N}}(x_0^*))]$
    Test Stopping Conditions: optimum design point against previous design
    if Stopping Conditions $\neq$ true **then** ▷ Step 4
        Update Surrogate models:$\tilde{f}_i(x_{0i}, y_i)$ and $\tilde{c}_i(x_{0i}, y_i)$
    else Accept Surrogate solutions: $y_{1\cdots N} = \hat{y_{1\cdots N}}(x_0^*))$
until $13 \Rightarrow 26$: Stopping Conditions = true
return Converged solution: $[v^*] = [x^*, x_0^* y_{1,\cdots,N}^*]$

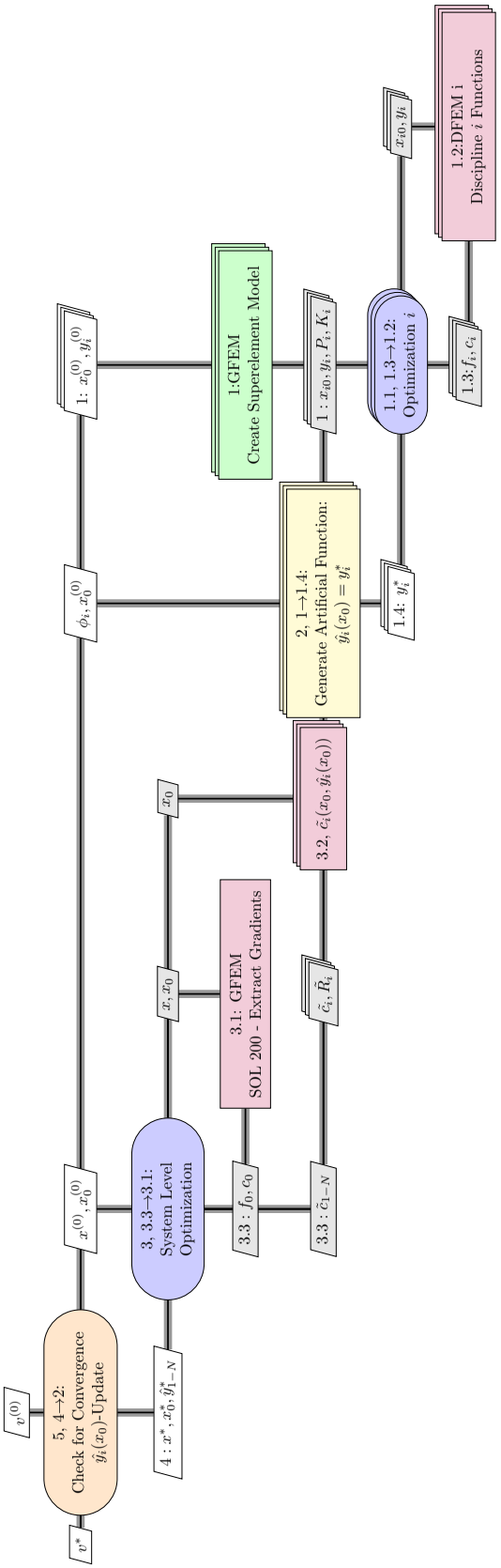


FIGURE 5.1. Diagram for the Primal Optimisation procedure

5.2 Alternating Direction Method of Multipliers

The ADMM procedure creates an aggregate Lagrangian function, $\bar{L_p}$ in equation (5.5), of the multi-model problem. The resulting function is then solved in a phased approach, iterating between 'leader', 'follower', and slack problems. In the case of hierarchical design the system-level optimisation problem is the leader, and subsystem-level problems are the followers.

$$\bar{L_p}(x,x_0,y,z,\mu) = F_0(x,x_0) + \sum_{i=1}^{N} w_i.f_i(y_i) + \sum_{i=1}^{N} \{\mu_i(\bar{c_i}(x_0,y_i) + z_i^2) + \frac{w_i}{2}|(\bar{c_i}(x_0,y_i) + z_i^2)|^2\} \tag{5.5}$$

Creating the aggregated function included normalising the objective responses to a similar order of magnitude as the sum of local constraint functions. ADMM uses an outer loop update method to rescale Lagrangian multipliers, but including the normalisation meant that the first design loop would at least have a similar problem scaling between different functions.

The procedure can be summarised as follows.

Step 1: Minimise the unconstrained aggregate function with respect to slack variables, z.
Step 2: Minimise the constrained aggregate function for system and linking variables, $[x,x_0]$.
Step 3: Minimise local subproblem with respect to local design variables, $y_{1\cdots N}$
Step 4: If a steady state solution has been reached, the optimisation has converged, otherwise update Lagrangian and weighted multipliers.

5.2.1 Multi-phase optimisation

Slack Variable Optimisation: Slack variables are used to convert subproblem inequality constraints into equality constraints. For some starting position, $v^{(0)}$, the constraint and objective function responses, F_0, $f_{1\cdots N}$ and c_0, $\bar{c}_{1\cdots N}$ remain constant, with respect to z. The resulting unconstrained non-convex problem can be solved using a GA, with gradient-based local optimisation: `fminunc`. The optimised value, z^{k+1}, is given by equation (5.6), to be used in the next phase of the optimisation problem.

$$\boldsymbol{z}^{k+1} = arg\min_{z} |\bar{L_p}| \tag{5.6}$$

Aggregate Function Optimisation: The function $\bar{L_p}(x,x_0)$ represents the leader problem. The Lagrangian form of the subproblem function and constraint responses are used to approximate the followers response to the leader's actions solving for a solution to equation (5.7).

$$[\boldsymbol{x},\boldsymbol{x_0}]^{k+1} = arg\min_{\boldsymbol{x},\boldsymbol{x_0}} \{\bar{L_p} : c_0(\boldsymbol{x},\boldsymbol{x_0})) \leq 0\} \tag{5.7}$$

Local subproblem Optimisation: The final optimisation phase solves for the follower function problem, solving for subsystem-level design variables, y^{k+1}. The follower optimisation problem simplifies to the constrained subsystem-level problem, equation (5.8).

$$y^{k+1} = arg \min_{y_i} \{f_i(y_i) : \bar{c_i}(y_i) \leq 0 \in i = 1,...N\} \tag{5.8}$$

5.2.2 Multiplier Update

The weighted penalty multiplier, $w_{1...N}$, represents the importance of subproblem constraint tolerance. This is updated through a ramp function similar to that used in the ATC formulation, equation (4.12). Penalty increases are based on subproblem design constraint violations, equation (5.9).

The multiplier weighting gradually increases the focus on subproblem functions with violated constraint functions, based on ramp rate, β = 4, from equation (5.9).

$$\boldsymbol{w_i}^k + 1 = \begin{cases} w_i^k & \text{if} \quad |\bar{c_i}|^k \leq 0 \\ \beta w_i^k & \text{if} \quad |\bar{c_i}|^k > 0 \end{cases} \quad \text{for } i = 1,...N \tag{5.9}$$

The Lagrangian multiplier $\mu_{1...N}$ represents the objective function ratio of function response between system and subsystem level problems. The Lagrangian gradient step uses the method of multipliers, equation (5.10). Step size α = 2 was used, which can yield satisfactory performance under a range of conditions [123].

$$\mu_i^{k+1} = \mu_i^k + \alpha . w_i(\bar{c_i}(\boldsymbol{x_0}, \boldsymbol{y_i}) + z_i^2) \tag{5.10}$$

Stopping Conditions: ADMM is an open feasibility strategy with regards to local constraint satisfaction with added penalty weightings. The convergence properties will be based on when a steady state solution, v^k, is reached and the optimisation problem has local constraint feasibility, $\bar{c_{1...N}}^k$. This is shown in equation (5.11), where ϵ represents a very small feasibility tolerance.

$$\begin{aligned} &||\boldsymbol{v}^k - \boldsymbol{v}^{k-1}||_\infty < \epsilon \\ &max([\boldsymbol{c_0}, \bar{\boldsymbol{c_{1...N}}}^k]) < \epsilon \end{aligned} \tag{5.11}$$

5.2.3 ADMM Process Description

Algorithm 5 ADMM Optimisation Procedure

Inputs Starting position: $[v^{(0)}]$, Bounds: v^{L}, v^{U}
Set initial weights, $\boldsymbol{w}^{(0)}$ and $\boldsymbol{\mu}^{(0)}$ and slack $[z^{(0)}]$;
repeat 3 $\Rightarrow$ 22
    function GA $\Rightarrow$ FMINUNC($z^{(0)}$); ▷ Step 1
        Aggregate function model: $\bar{L_p}(x^{(0)}, x_0^{(0)}, y^{(0)}, z^k, \mu^{(0)})$;
    return Partial solution: $[z^*] = [z^{k+1}]$
    function FMINCON($x^{(0)}, x_0^{(0)}$,[$x^{\mathrm{L}}, x_0^{\mathrm{L}}$[, [$x^{\mathrm{U}}, x_0^{\mathrm{U}}$]); ▷ Step 2
        System model: run GFEM `SOL 200` $\models F(x, x_0)$ and $c_0(x, x_0)$;
        parfor $i = 1 \rightarrow N$ **do**
            Subsystem model: run DFEM `SOL 200` $\models f_i(x_0)$ and $\bar{c}_i(x_0)$
        end parfor
        Aggregate function model: $\bar{L_p}(x^k, x_0^k, y^{(0)}, z^*, \mu^{(0)})$;
    return Partial solution: $[x^*, x_0^*] = [x^{k+1}, x_0^{k+1}]$;
    parfor $i = 1 \Rightarrow N$ **do** ▷ Step 3
        function FMINCON($[y_i^{(0)}]$, y_i^{L}, y_i^{U});
            Subsystem model: run DFEM `SOL 200`, $\models f_i(y_i)$ and $c_i(y_i)$;
        return Partial solution: $[y_i^*] = [y_i^{k+1}]$;
    end parfor
    Concatenate Distributed Variables: $y^* = [y^{k+1}]$, $\bar{c}^*_{1\cdots N}$;
    Test Stopping Conditions: System-Subsystem Converged Feasible ▷ Step 4
    if Stopping Conditions $\neq$ true **then**
        Update Penalty weightings and Lagrange Multipliers $\boldsymbol{w}^{k+1}, \boldsymbol{\mu}^{k+1}$
        Update Starting Positions: $v^{(0)} = [x*, x_0^*, y^*_{1\cdots N}]$
until 3 $\Rightarrow$ 22: Stopping Conditions = true
return Converged solution: $v^* = [x^*, x_0^* y^*_{1,\cdots,N}]$

5.3 Primal Distributed Optimisation Results

5.3.1 Surrogate BLISS Optimisation Results

BLISS is a nested primal decomposition strategy. The version presented here uses a surrogate based system-subsystem approximate interaction model to produce locally optimal subsystem designs with respect to system-level design inputs. A SAGO was used to first generate the surrogates and an approximate solution at the current design point. The quality of the final subsystem design solution is dependent on the quality of the surrogates produced.

Two surrogate methods were tested: a gradient-enhanced kriging method and a kriging response surface model with Gaussian correlation function and second order regression. The former, although faster, could not produce stable surrogates which could be reflexively updated throughout the design, so the final surrogate used the kriging response surface model. The oldest and least accurate sample points in were also periodically removed in-order to best reflect the solution space at the current design point.

Figure 5.2(a) shows the objective mass against the cumulative FE evaluations. Figure 5.2(b)-(d) shows the nested function evaluations for the system-level GFEM, subsystem-level DFEM and surrogate solutions, with respect to outer loop iteration.

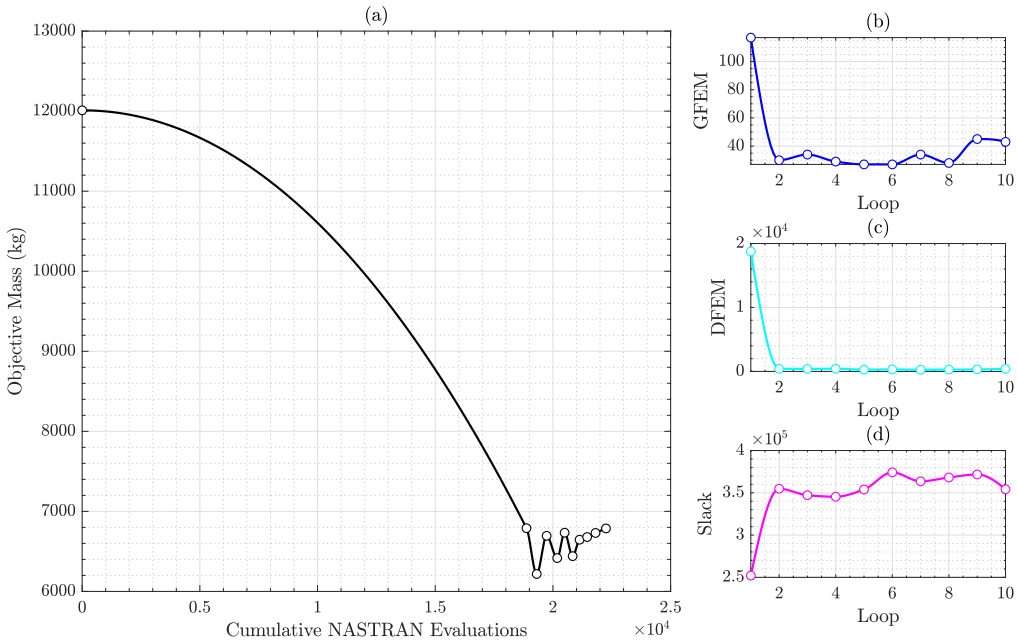


FIGURE 5.2. BLISS: Optimistic Outer Loop Optimisation Results Summary; (a) Objective Mass - Function Evaluations, (b) GFEM Evaluations, (c) DFEM Evaluations, (d) Surrogate Evaluations

Surrogate supported models have a high start-up cost for generating the initial surrogates. However, the rest of the analysis has comparatively much fewer evaluations than other distributed methods tested, such as ATC. Effectively expending computation initially rather than metering it over several iterations, plus the resulting surrogates could be used for further analysis later in the design process. During this analysis, surrogate evaluations were exceedingly high, nested functions using a non-surrogate approach would have been prohibitively computationally expensive, especially when the subsystem is of similar effective cost as the system-level model.

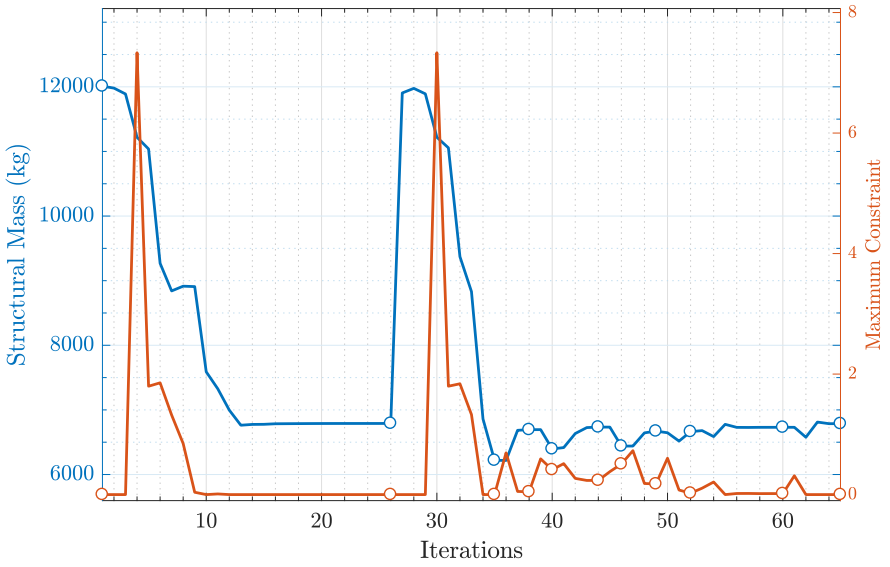
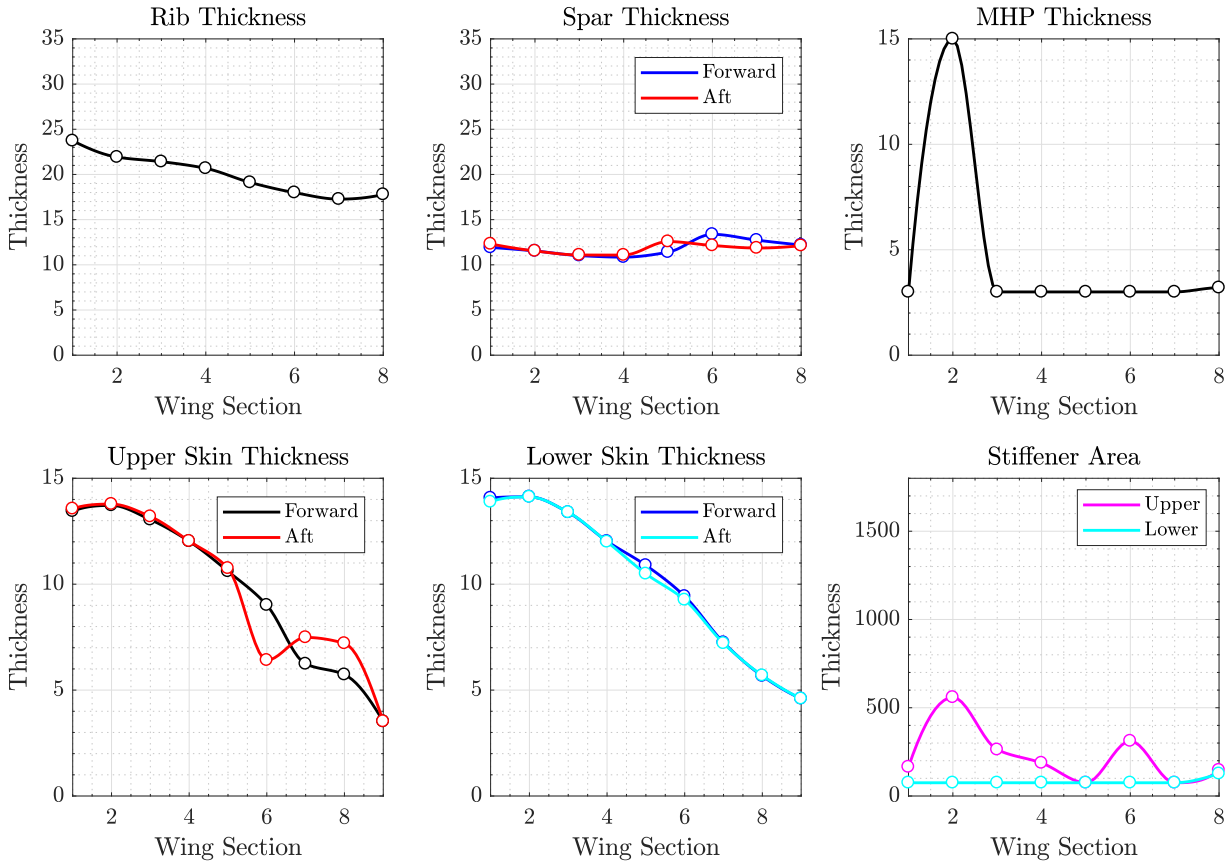


FIGURE 5.3. BLISS: Optimistic Problem Objective Function Iteration History; Satisfactory MaxC and final optimised objective mass, $M^* = 6786.81$ kg.

Figure 5.3 shows the design iteration history for the full model design mass and constraints. The final solution is $M^* = 6786.81$ kg, 19% higher than the optimistic AAO solution. The final design meets all given constraints but is significantly different from previous solutions. The disparity between final designs could the result of limitations with the surrogate model. All local solutions are dependent on the accuracy of concurrent surrogate models producing coherent results. If the one or more of the local subproblem solution spaces substantially changes relative to external variables throughout the analysis, simply updating surrogates with the new responses may lead to errors. For particularly variable solution spaces it may be necessary to regenerate surrogates entirely. However nested surrogate regeneration may result in exceedingly high computational expense periodically slowing down the algorithm.

Figure 5.4 shows the design variable summary and constraint value distribution for the optimistic solution. The design features sharp inconsistent local variations, particularly the upper skin, MHP and stiffener areas. The constraint distribution also indicates that some constraints are over-designed, especially when compared to previous results.

(a) Design Variable Breakdown



(b) Constraint Value Distribution

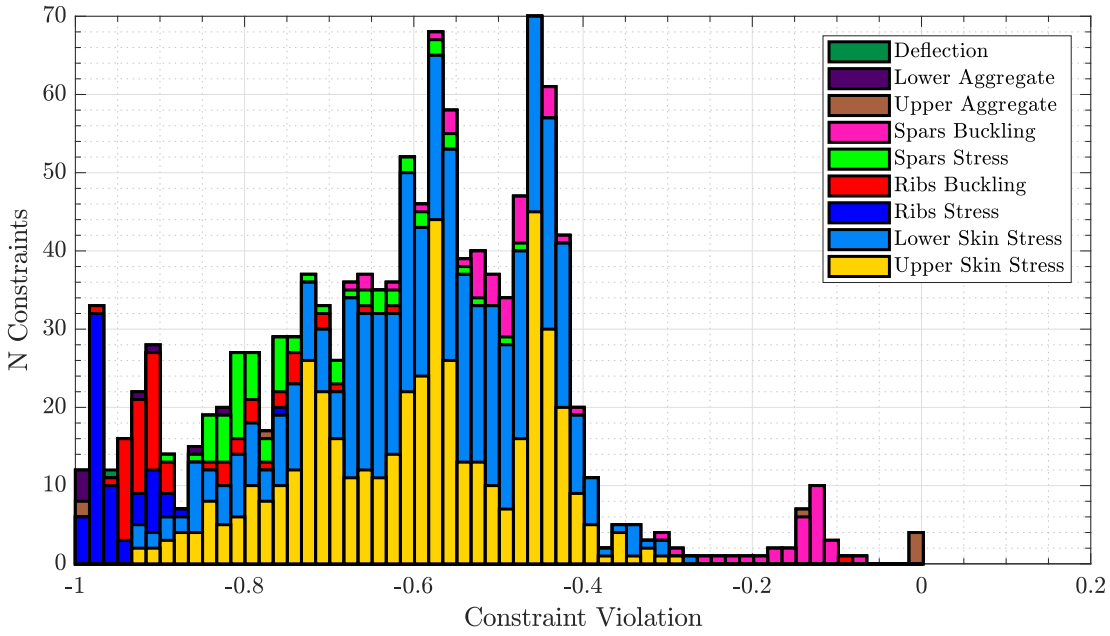


FIGURE 5.4. BLISS: Optimistic Problem Final Design Summary, $M^* = 6786.81$ kg.

5.3.2 ADMM Optimisation Results

ADMM is a multi-phase alternating optimisation architecture, solving the aggregate slack coupling problem, system and distributed subsystem-level problems in turn, with an outer loop Lagrangian update. Figure 5.5(a) shows the objective mass against the cumulative FE evaluations. Unlike BLISS, the ADMM method has a more consistent number of subsystem function evaluations throughout the analysis process.

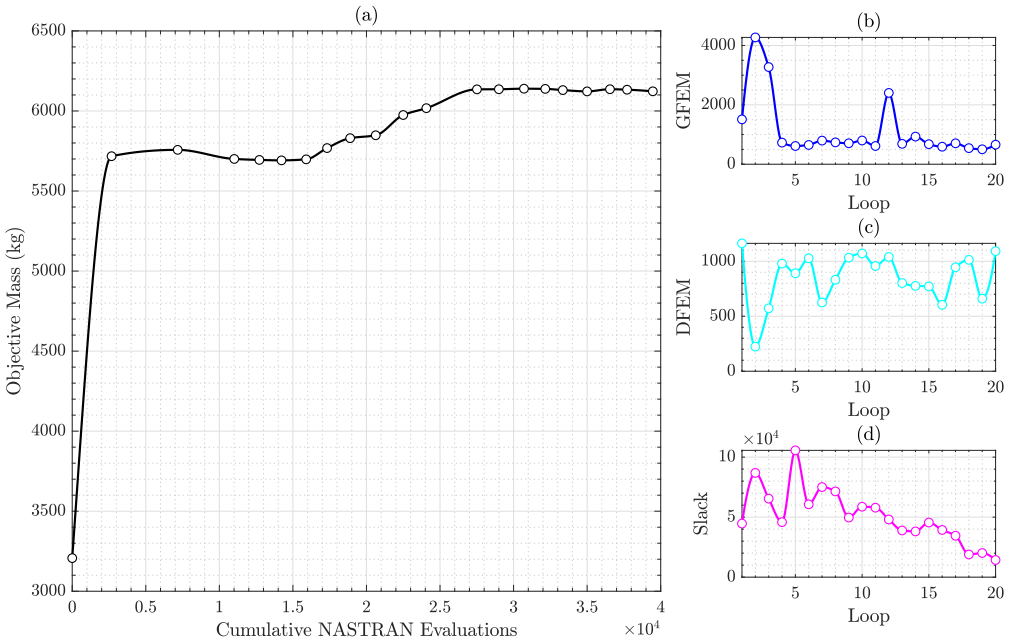


FIGURE 5.5. ADMM: Optimistic Outer Loop Optimisation Results Summary; (a) Objective Mass - Function Evaluations, (b) GFEM Evaluations, (c) DFEM Evaluations, (d) Surrogate Evaluations

Figures 5.2(b)-(d) show the iterations of each phase of the optimisation process with respect to outer loop updates. The slack coupling optimisation problem for the aggregate function does not require FE model evaluations. Similar to the augmented ATC procedure, ADMM more effectively spreads the computational requirements across the entirety of the optimisation process.

Figure 5.6 shows the objective-constraint iteration history for the optimistic design problem, with outer ramp rate $\beta = 4$. Figure 5.7 is the final design and constraint breakdown for the ADMM procedure. Table 5.1 summarises a set of parameter studies, between the optimistic and pessimistic assumptions, with a variable ramp rate.

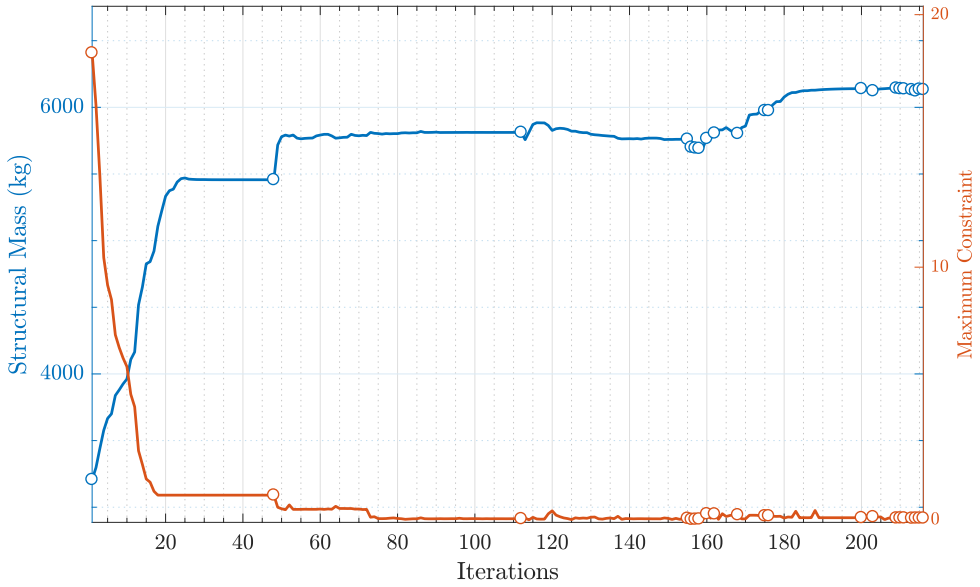


FIGURE 5.6. ADMM: Optimistic Problem Objective Function Iteration History; Final active constraint = 0.0606 and final optimised objective mass, $M^* = 6122.71$ kg.

TABLE 5.1. ADMM: Parameter Study

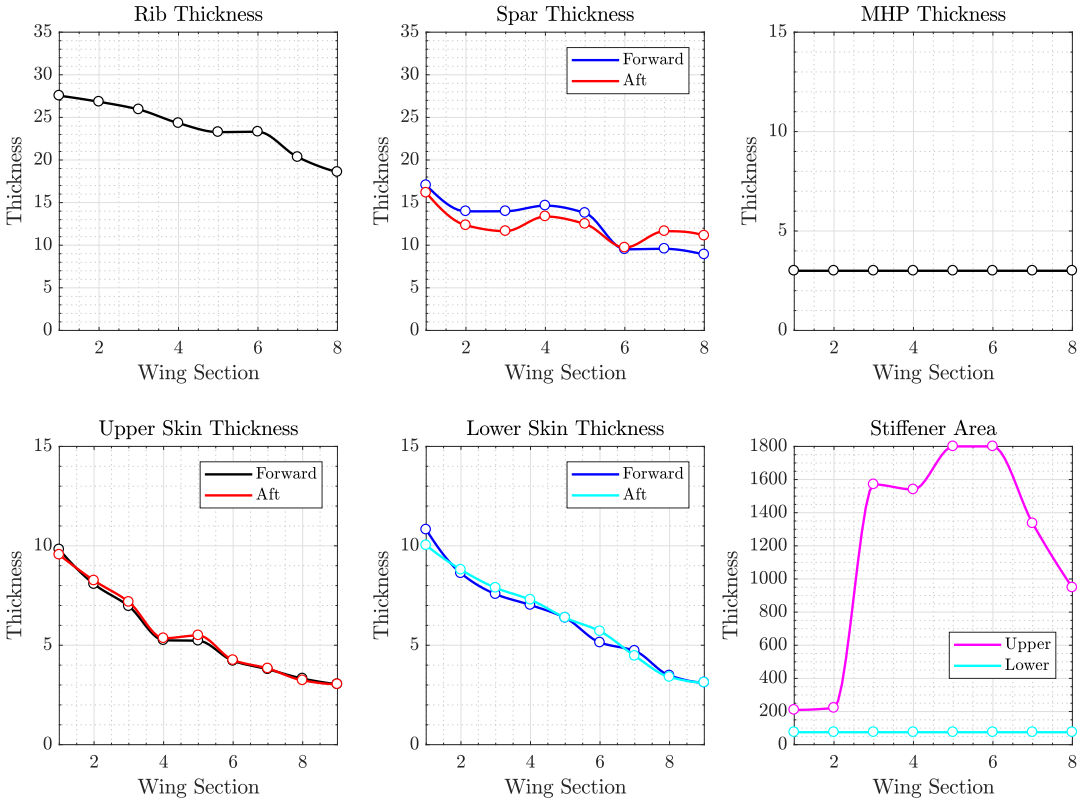
	Optim. Beta = 2	Optim. Beta = 4	Pess. Beta = 4
Objective Mass (kg)	6152.76 (%)	6132.89 (+7.27%)	6838.41 (+19.61%)
Objective Mass Variation[1]	+7.62%	+7.27%	+19.61%
MaxC	0.010	0.061	0.161
Outer Loops	16	20	6
Function Evaluations	27916	39465	18580

[1] = Mass Variation from monolithic solution, $M^* = 5717.07$.

The final design using ADMM is an improved solution over the BLISS procedure. The design features relatively smooth variations in skin thickness distribution. However, the stiffener, MHP and spar thickness' feature irregularities in comparison to previous solutions. However, the constraint distribution shows that the optimisation converged with several violated constraints, with significantly fewer active-constraints than the AAO and ATC algorithms. The optimiser fails to converge to any meaningful solution under pessimistic conditions.

Positively the optimiser does perform better and more consistently than the BLISS procedure. Similarly, the lack of significant tuning parameters means that it is reasonably stable.

(a) Design Variable Breakdown



(b) Constraint Value Distribution

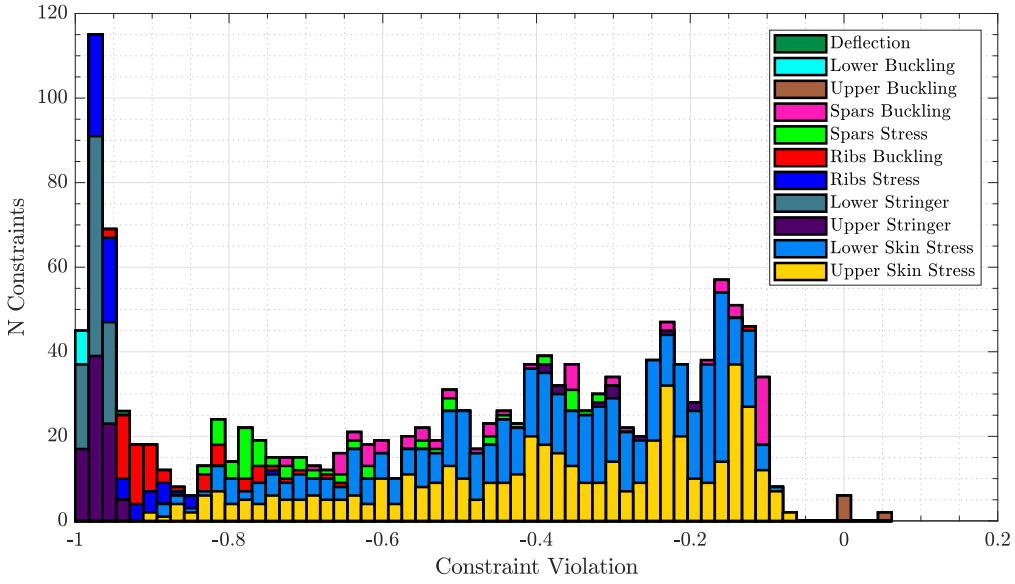


FIGURE 5.7. ADMM: Optimistic Problem Final Design Summary; $\beta = 4$, $M^* = 6122.71$ kg.

5.4 Chapter Summary

The primal distribution methods assessed were ADMM and surrogate based BLISS architectures. Primal methods remove the need for copies of linking variables in quasi-separable design problems or creating more extensive, potentially cumbersome problem statements. BLISS uses nested surrogate models to approximate system-subsystem model interactions. ADMM uses an alternating system-subsystem optimisation with respect to an aggregate Lagrangian problem statement.

Most combinations of state-parameters and starting position using primal distributed solvers led to an off-optimal result. Both the BLISS and ADMM procedures produce sub-optimal final designs significantly different from the monolithic or dual distributed methods. Primal distributed methods have been shown within the literature to produce good results for a range of problems. However, for the optimisation problem presented here two primal MLO architectures which could not produce viable results in comparison to the others tested here.

During tests on dual distributed procedures, it was found that a gradient-based selection method for linking variables improved the final design. In the case of primal methods, the local subproblems feature an even more limited design space than those tested in the ATC procedure. The difference between the partitioned model response and the primal model response meant that local subproblems were severely limited in solution space. The local optimisation subproblem, solved with respect to local variables alone, y_i, does not have the design range applicable to solve the problem presented adequately. Despite being sophisticated, the coordination strategies in both cases failed to synergise to represent the complete multi-model design adequately.

The next chapter presents a comparison of the monolithic, dual and primal distributed results, with a discussion of design metrics to better trade-off the use of one model or another.

CHAPTER 6

Optimisation Comparison & Discussion

This chapter compiles the results from the monolithic, dual and primal optimisation procedures for a given test problem for comparison. A range of quantitative mesures are used to identify the efficiency, accuracy and robustness of each optimiser. The best algorithm presented were then tested on a representative wing model under aeroelastic loading with a final comparison of solutions.

This thesis began with identifying the significant subsets of algorithm within the MLO literature as a range of monolithic, dual and primal distributed optimisation procedures. Each procedure was tested for a baseline multilevel problem. The objective was to minimise the structural mass of a simple wing-box structure, with both system constraints and subsystem constraints. Test problems for each optimiser featured an optimistic and a pessimistic design case, creating either a favourable or competitive system: subsystem relationship, respectively.

Figure 6.1 and table 6 summarises the iteration histories and final design values from each procedure for the optimistic version of the design problem. Table 6.2 shows the optimisation results from the pessimistic design problem.

TABLE 6.1. Summary of Optimisation Results for the Optimistic Problem

	Objective Mass (kg)	MaxC	Function Evaluations
AAO	5717.07	0.000	5321
SEQ	5601.41	0.000	20954
ATC	5478.46	0.003	24681
ADMM	6132.89	0.061	39466
BLISS	6786.81	0.000	203937

TABLE 6.2. Summary of Optimisation Results for the Pessimistic Problem

	Objective Mass (kg)	MaxC	Function Evaluations
AAO	5925.00	-0.004	11271
SEQ	5716.77	0.084	17306
ATC	5479.20	0.008	21838
ADMM	6660.91	0	18580

The best performing algorithms were the monolithic AAO and dual distributed ATC. The AAO solution required several attempts before finding a viable solution, the repeated function evaluations have not been taken into account in either table 6 or 6.2, as the number of start points and selection of the initial design points were not investigated as a part of this study. The ATC solution used a formulation augmented with Lagrange multipliers and gradient-based partitioning, with additional tuning of state parameters the best design solution across all the tested architectures was found.

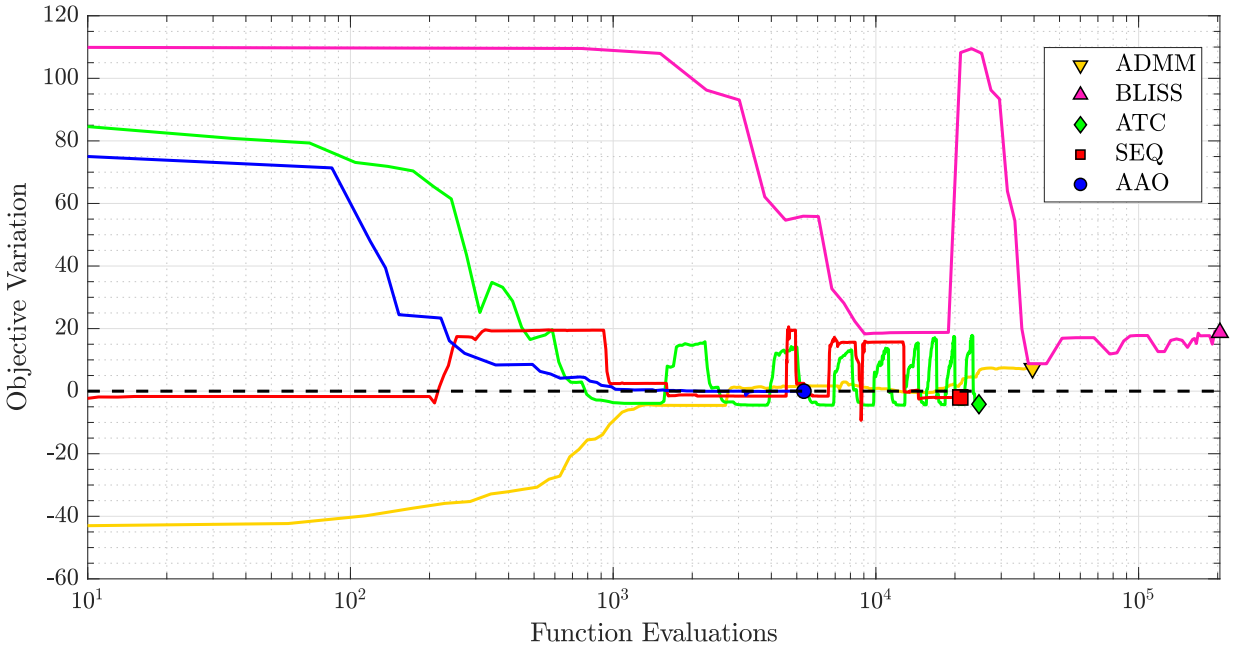


FIGURE 6.1. Optimisation Iteration Histories for the Optimistic Problem; Function evaluations vs. Mass Variation from monolithic solution, $M^* = 5717.07$, for monolithic, dual and primal distributed methods.

6.1 Monolithic vs Distributed Optimisation

The first distinction between methods tested was between monolithic and distributed methods.

Monolithic architectures attempt to solve the multi-model problem directly concatenating all wing and sub-components features into a single design step. The optimisation tested were the single-phase: 'all-at-once' (AAO) optimisation method, and the multi-phase: sequential level mixing (SEQ) optimisation method.

Distributed methods decompose the original problem into a set of smaller isolated sub-problems which are then subject to additional coordination requirements in a multi-phase algorithm. The methods tested were the dual distributed method: analytical target cascading (ATC), and the primal distributed methods: bi-level integrated system synthesis (BLISS) and alternating direction method of multipliers (ADMM).

The monolithic procedure was the most efficient tested by a significant margin. The AAO procedure converged with the fewest evaluations compared to the other methods presented. The next most efficient solver was the SEQ method, using multiple starting positions at each level contributing to the increased computation. The AAO procedure took ~ 25% of the function evaluations used in the sequential analysis, ~ 20% of those used in the dual distributed ATC procedure, and less than 15% of those for the primal distributed methods.

Monolithic solvers favour the overall system, featuring feasible, but off-optimal substructures. The result is that optimistic design problems are simpler and more effective to solve using monolithic methods. When applied to uncertain MLO design problems, the solution space for satisfactory designs is uncertain, and it becomes difficult to assume monolithic methods will provide the most optimal solutions.

The ATC procedure produced the best overall designs in both the optimistic and pessimistic case, more so in the pessimistic design case where the best AAO solutions varied significantly with starting position. ATC is a top-down approach to optimisation. Increasing the relative difference between system: subsystem-level optima reduces the outer loop convergence rate, reducing algorithm efficiency. In simpler terms, the more pessimism in the design assumptions, the more aggressive the outer loop update strategy must be to maintain efficiency.

Distributed solvers in general, have increased algorithmic complexity, improve robustness to variation in the model relationship. However, the overlying algorithms feature additional state-parameters which require tuning for the specific problem at hand. Tuning solvers accurately is a non-trivial problem as the relationship between state-parameters and model optimiser response is not always apparent, and yet poor tuning can lead to significantly different results.

Monolithic solvers, in their simplicity, allow for a more general, improved resulting design, rather than the best optima within the design space. The simpler the method, and the fewer assumptions needed about the problem, the more general a problem approach can be taken.

6.2 Meaningful Decomposition & Partition

Decomposition is the process of modelling a system through a series of smaller components, parts or substructures. Hierarchical models, like those used in structural design, are readily expressed as a series of localised subsystems. The entire design, although made up of a multi-model system is expressed as the collection of design variables and responses at all levels of design.

Partitioning is the process of splitting a collection of design variables and responses into a phased or distributed problem with smaller, simpler subproblems to produce a practical result. The ideal partition retains and accurately depicts the original design problem while making each subproblem more straightforward to solve.

MLO within structural design involves the decomposition of the structural model and partitioning the disparate models into a series of determinate subproblems. This process is meaningful when it accurately depicts the original non-parsed design problem while allowing for smaller, more numerical determinant solutions to subproblems to produce a practical result.

The performance range defines all the possible solutions between the upper & lower bounds of the design problem. Figure 6.2 shows the probability density function (PDF) for the upper and lower root wing panel substructures, outlining the objective performance range of each.

A PDF is made using a space-filling sampling algorithm such as LHS to define a range of

possible points in the design space. The sampled points are tested until the cumulative mean of responses become consistent and unchanging, indicating that the sampled solution space is representative of the real model responses. The result is a normally distributed performance range about the mean response [148]. The PDF depicted in figure 6.2 are bounded by the upper and lower bound of design variables.

The 'local' objective range, is based on a design problem only using the local panel design variables $[y_i]$. The 'dual' objective range uses the physical panel-based selection matrices for linking variables, combined with the local variables $[x_{0i}, y_i]$.

(a) Upper Wing Panel Model - Root

(b) Lower Wing MHP Model - Root

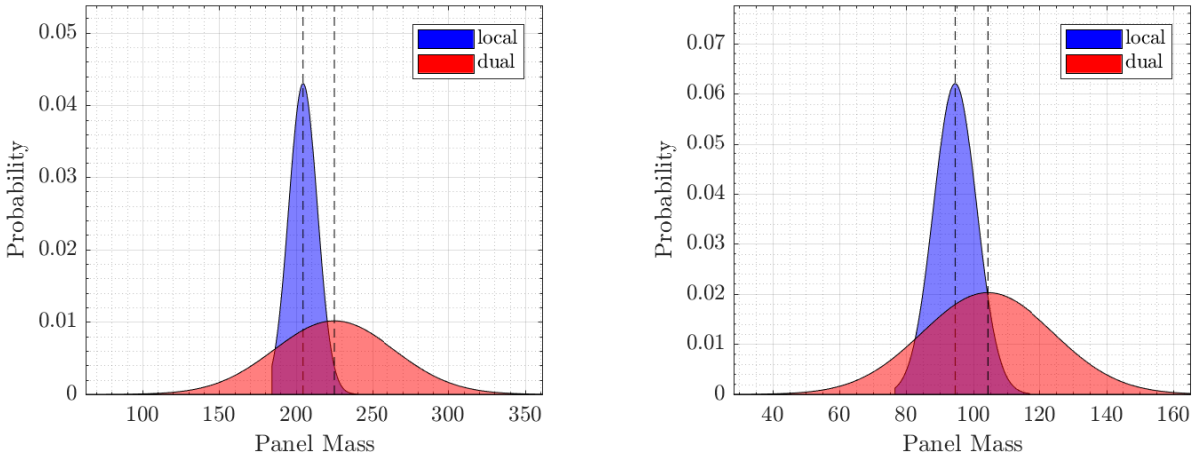


FIGURE 6.2. Subproblem Objective Function Design Range PDF; Based on a space filling Latin hyper-cube sampling of each design problem.

The design range from the local partition is much narrower than the dual partition. In both the upper and lower panel subproblems, including the local linking variables more than doubles the available design range of the optimisation problem.

Distributed solvers produce designs featuring locally optimal substructures with a feasible systems level design. Both partitioning strategies produce optimal solutions to fundamentally different problems. In the local partitioning case, the design found may be optimal for the given objective performance range but widening the partition to include more linking variables expands the solution space available to an optimiser.

Lagrangian multipliers, outer loop updates and penalties are used in both the ATC and ADMM procedure; they alter the local problem solution space so globally optimal MLO solutions exists within the feasible subproblem design range. However, if this performance range is too narrow, changes too much with external variables, or does not capture the original problem, any external design space alterations only increase the number of function evaluations to produce an off-optimal result.

6.2.1 Information Certainty

Product development metrics are used to measure the extent to which a systems design parameter is known at a point in the design timeline [149]. Within MLO, this can be used to measure how meaningful any partitioned design subproblem is to the associated design objectives and constraints. Information Certainty (I_C) is calculated using equation (6.1), containing the ratio of PR_i^k, the *performance range* of the current iteration, k, for subproblem i of N total subproblems, against the range of the initial complete monolithic problem. The performance range is the difference between the maximum and minimum possible response, representing the breadth of solutions possible.

$$I_C = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{\mathrm{PR}_i^k}{\mathrm{PR}_i^{(0)}}\right) \tag{6.1}$$

TABLE 6.3. Information Certainty Variation with Partition Strategy

	$f(x)$	$\bar{c}_{opt}(x)$	$\bar{c}_{pes}(x)$
local	0.240	0.622	0.261
dual	0.956	0.816	0.982
gradient	1.000	0.954†	1.000
monolithic	1.000	1.000	1.000

† = Gradient-based partition only only captures the most dominant design variables within some order of magnitude tolerance of the mean gradient.

In MLO, I_C can be used to measure the quality of subproblem partitioning. Table (6.3) summarises the I_C for a range of partition schemes. The local panel partitions have the lowest quality partitions, only able to simulate ~ 25% of the solution space. Note, between the optimistic and pessimistic problems the quality of partitions changes.

A narrower I_C implies the partitioned design problem captures little of the substructure solution space. This lowers the chances of the MLO global optimum being found within each subsequent optimisation and requires a more robust outer loop coordination amongst subproblems. A broader I_C indicates a larger solution space, capturing much of the subproblem behaviour. Problem efficiency has been shown to improve most by reducing the number of outer coordination updates, even when increasing the number of linking variables.

6.3 Pessimism in Subproblem Relationships

When optimising multi-model systems, the quality of the solution and applicability of an optimiser is determined by the decomposition and inter-model interactions. In cooperative DSP (Decision-Support Problem) models, it can be trivial to determine when problems are *optimistic* or *pessimistic*. In multi-model design systems this relationship is not always as clear. The labels used in the research of optimism and pessimism were identified as one problem features assumptions which narrow the overlapping feasible solution space amongst subproblems. The result being an increased level of inter-model competition.

Figure (6.3) shows an exemplar constraint function response PDF. The major features of the constraint PDF will help define pessimistic design assumptions; features outlined includes:

- *Performance Range* (PR): As before, the maximum range of possible responses available in the current design problem.
- *Target Range* (TR): The area under the pdf marked satisfied (with constraint values less than 0), is defined as the target range. The area which represents the probability any given solution satisfies constraints.
- *Lower Requirement Limit* LRL: Defined as the area between the mean value in the solutions space, μ, and the lower bound. This area represents the overall system design capability, and the general spread of data across the solution space.

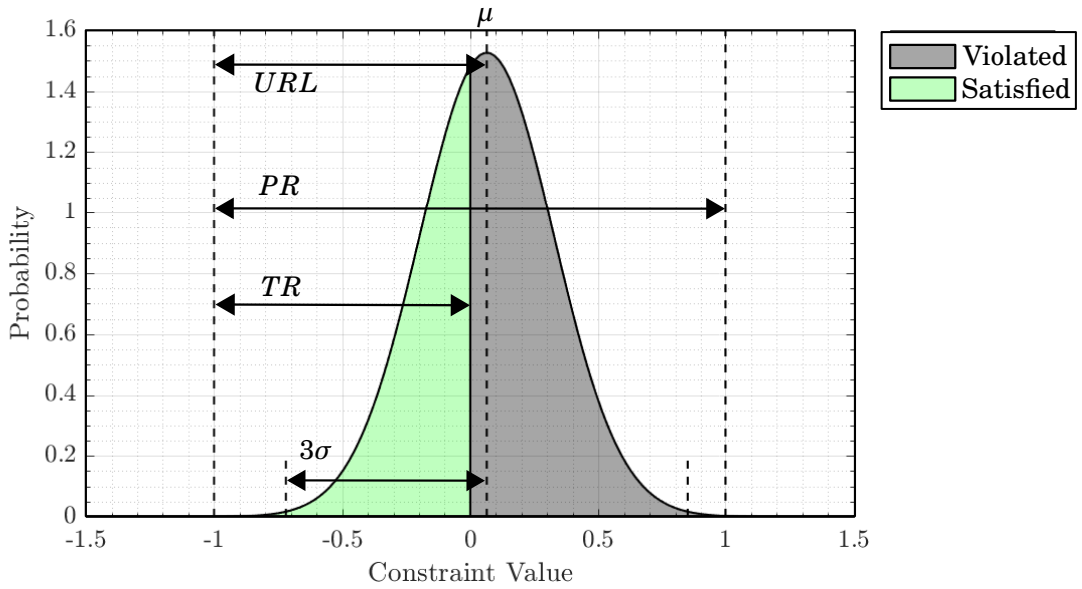


FIGURE 6.3. Constraint Response PDF metrics; Lower Root Panel with Pessimistic problem

By combining solution metrics with observations between the two design positions identified, it is possible to define a range of indicators to best inform optimisation design. Figure 6.4 shows the constraint response PDFs for the upper and lower root panel substructures, under both optimistic and pessimistic positions, with a range of decomposition strategies.

The pessimistic positions feature a greater number of stress constraints and shows that the solution space shifts to feature more infeasible designs, where constraints are more likely to be violated (> 0). This is because the active constraint across the entirety of the solution space change from buckling, to stress constraints under different design assumptions.

Narrowing I_C magnifies the differences between the optimistic and pessimistic solution spaces, as shown in figure 6.4(f), where the solution spaces have no common solutions. For the upper panel figures 6.4(a,c,e) the performance range is more consistent under most types of decomposition. The performance range is more stable because the upper panel subproblem is a detailed part-level structure of an *existing* component. Meanwhile the lower panel is a detailed non-generic substructure which does not exist in the system-level model, and has significantly higher stresses about the rim of the MHP.

6.3.1 Design Freedom

Design Freedom (D_f) is defined as the extent to which system can be "adjusted" while still meeting design requirements. The equation for D_f is given in equation (6.2), measured by the overlap between the the target range and system performance normalised over the achievable performance range [149].

$$D_{fi} = \frac{\mathrm{TR}_i \cap \mathrm{PR}_i^k}{\mathrm{PR}_i^{(0)}} \text{ for } i = 1 \cdots N \tag{6.2}$$

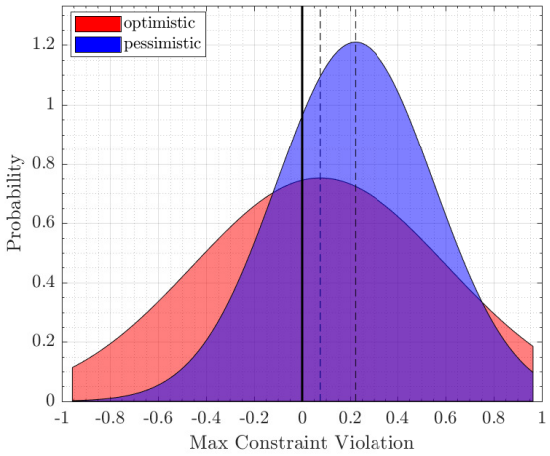
TABLE 6.4. Design Freedom Variation with Subproblem; D_f for subproblem using panel-based partitioning

	Upper Panels		Lower Panels	
	D_f opt.	D_f pes.	D_f opt.	D_f pes.
Root	0.496	0.295	0.718	0.182
Mid	0.517	0.434	1.000	0.476
Tip	0.621	0.581	1.000	0.979

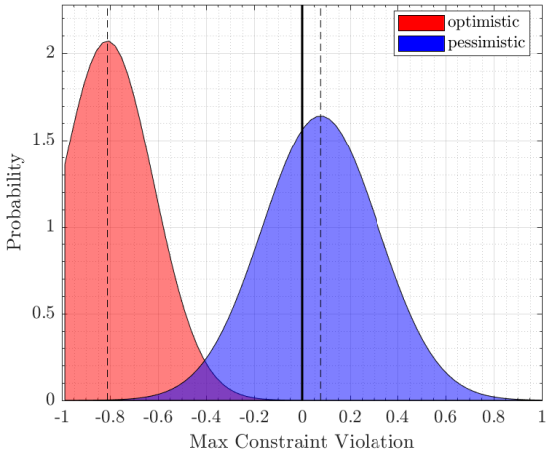
Measuring design freedom in MLO indicates the problem difficulty of constraints for a given partition within subproblems. Table 6.4 summarises the design freedom for each type of parition. The narrower the D_f, the more difficult the design subproblem. Designers can increase a partition to widen the solution space, or ensure a more numerically robust local optimisation algorithm and coordination strategy for these subproblems. A broader D_f indicates a better partition and a more "solvable" subproblem.

The D_f values between the optimistic and pessimistic upper-panel problems are consistent with the observations based on the constraint PDFs, figures 6.4. The upper panel D_f does narrow between the two design positions, but the solution space is consistent. Whereas for the lower panels there is significant variations between the two design positions. For the optimistic design case most of the lower-panel subproblems feature design feasible solution spaces, which are made infeasible in the pessimistic case. When design subproblems feature detailed non-generic substructures with significantly more difficult design constraints, a narrower D_f, it is more likely the resulting design is pessimistic.

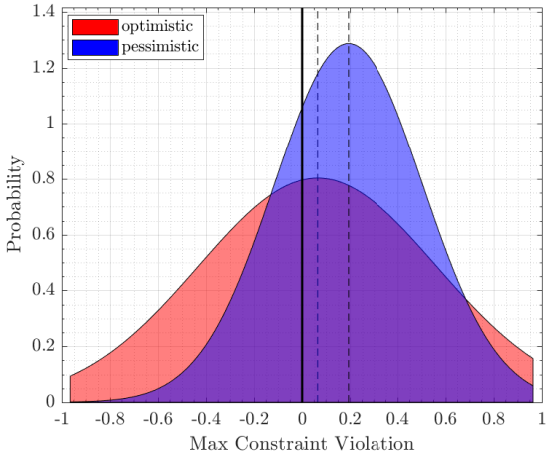
(a) Monolithic - Upper Root Panel



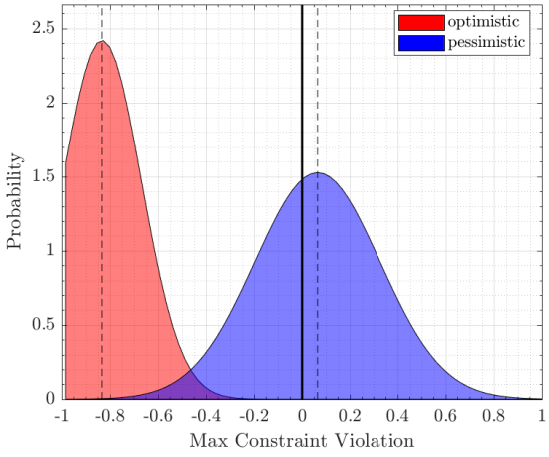
(b) Monolithic - Lower Root Panel



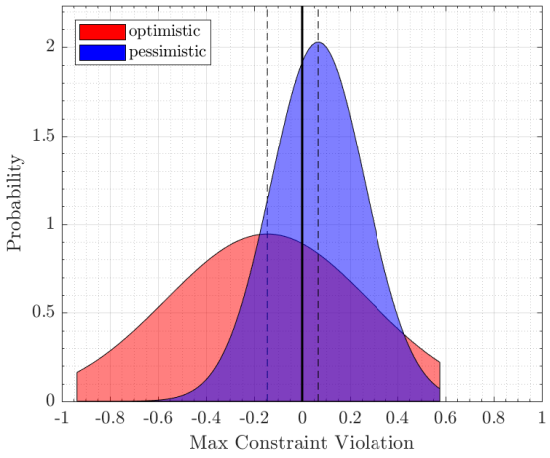
(c) Dual - Upper Root Panel



(d) Dual - Lower Root Panel



(e) Local - Upper Root Panel



(f) Local - Lower Root Panel

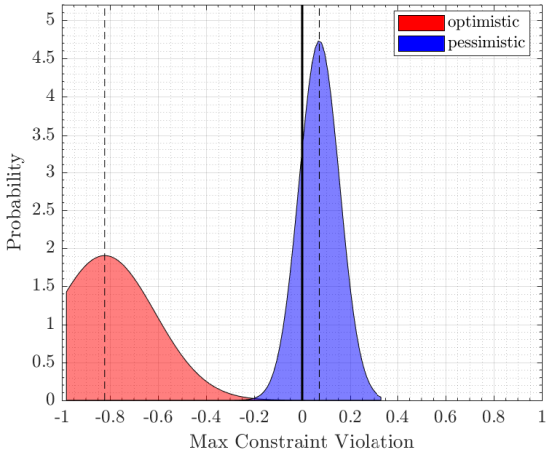


FIGURE 6.4. Constraint Design Range for Root Panel Substructure; under differ net design assumptions & Decomposition

6.4 Solution Space Constraint Screening

6.4.1 Solution Space Convexity

The local substructure buckling modes responses were non-convex; featuring discrete changes in the solution space as buckling modes shift with respect to design variables. Aggregating constraints and testing for multiple start positions was the only way to find adequate solutions to non-convex subproblems. In the optimistic monolithic AAO optimisation, a consistently convex solution space was apparent, indicating the multi-model appeared to be convex. However, upon inspection, the local designs for each solution for the optimistic AAO problem are substantially different from one another. The local design suffered as a result of coupling the non-convex local responses with the much larger convex responses from the system-level.

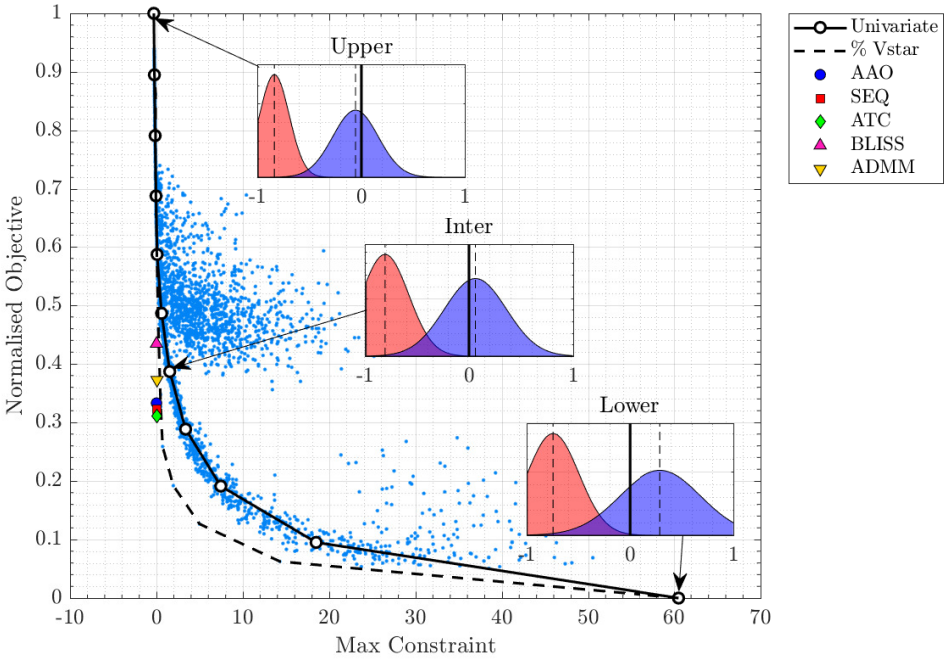


FIGURE 6.5. Normalised Mass:MaxC Polar; with overlay of solution space PDFs for the optimistic, and pessimistic design cases for the dual-lower root panel.

Figure 6.5 shows the normalised mass: maximum constraint polar of the AAO design problem. The sampling points were based on a Latin hyper-cube sampling; the dashed line showing response variation as a linear % of V^* for the optimistic AAO solution; the white markers indicate solution variation for a uni-variate distribution. The inlay figures show the PDF solution spaces for the dual decomposed optimistic and pessimistic problems (in red and blue), at three points in the design space: the upper-bound, lower-bound, and an intermediary point of the design space.

The design envelope in figure 6.5 shows the general trends across the AAO solution space, with local PDFs showing a representative subproblem distribution. The D_f for the optimistic problem is extremely broad across most of the solution space. The result being that local design variables, $y_{1\cdots N}$, MHP sizing and stiffener sizing can vary whilst still finding an "optimal" feasible design.

6.4.2 Constraint Scaling

Limited scaling between constraints in multi-model design problems leads to localised or non-active responses being neglected. The monolithic models used limited constraint scaling, the AAO optimiser neglected local variables and missed significant weight savings in the design space. The SEQ improved on the AAO solution, but the design variable breakdowns for local variables were still different from the better ATC solutions.

Figure 6.6 shows the normalised mass-maxC polar with subgroups from the system-level: upper skin stress, spar buckling and deflection, and lower panel buckling from the subsystem-level. Each polar is based on a set of sampling points from the AAO design problem, using the maximum constraint from each subgroup.

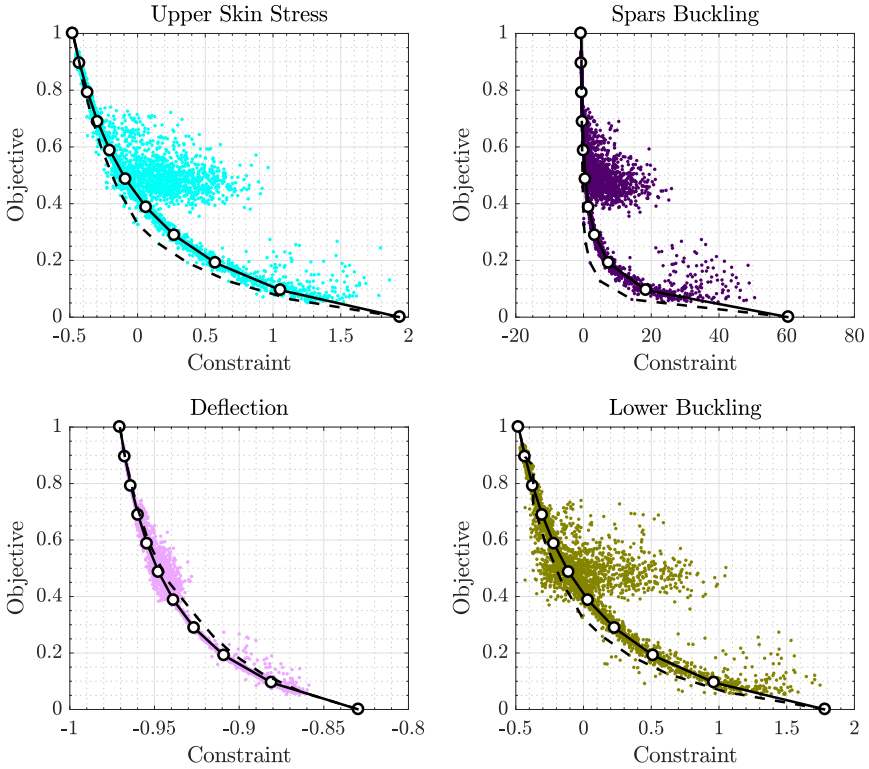


FIGURE 6.6. Normalised Mass: MaxC Polar; constraint response envelopes

The tip deflection constraint remains inactive throughout the solution space, with a much narrow solution range than the others presented. The maximum upper skin stress and lower panel buckling show a normal distribution across the constraint. Whereas the spar buckling is an order of magnitude larger than the other three constraints for most of the design space, whilst still being bounded at feasible max at lower bound -1.

In the optimistic design case, the D_f is so broad that a range of feasible but not optimal solutions can be found, resulting in a 'pseudo-convex' solution space. In the pessimistic design case, the active constraint moves from the system-level to the subsystem-level design, and the effects of poor constraint scaling become more apparent, creating the non-convex solution space.

6.4.3 Design Capability

Measured constraint scaling requires evaluation of the design solution range available. Design capability measures is the normalised measure of the range of possible function responses. When combined the mean, variance and design capability (μ, σ and C_d respectively) can be used to assess the breadth of constraint values available within any possible partition. These measures vary from the previous measure design freedom, D_f, as this does not take into account the feasible range, only the how wide the solution space is.

These measures cannot improve the problem of gradient scaling when combining designs with dominant features; it can only provide a better framework for comparison, and information on combining dissimilar design responses [150].

Equations (6.3) and (6.4) represent the normalised upper and lower spread of the function response, all possible solutions rest between these bounds. The minimum of the two, equation (6.5), represents the normalised range within the feasible system.

$$C_{dL} = \frac{\mu - \text{LRL}}{3\sigma} \tag{6.3}$$

$$C_{dU} = \frac{\text{URL} - \mu}{3\sigma} \tag{6.4}$$

$$C_d^k = \min\{C_{dL}, C_{dU}\} \tag{6.5}$$

The broader the design capability, C_d^k, for any iteration k, the larger the potential for design space exploration. Table 6.5 shows the mean (μ), standard deviation (σ), and the design capability (C_d) for the monolithic design problem, based on the sampling points from figures 6.5 and 6.6.

TABLE 6.5. Design Capability Variation with Select Constraint

	μ	σ	C_d
Upper Skin Stress	0.155	0.446	0.473
Lower Skin Stress	0.164	0.453	0.468
Ribs Buckling	1.054	2.482	0.265
Spars Buckling[‡]	5.469	8.143	0.253
Deflection[§]	-0.940	0.024	0.428
Upper Buckling	0.102	0.457	0.422
Lower Buckling	0.071	0.436	0.421
Upper Detailed Stress[¶]	0.274	0.527	0.571
Lower Detailed Stress[¶]	0.000	0.412	0.422

[‡] = Highest value solutions but lowest design capability; could be improved through re-scaling.
[§] = Solution range indicates this constraint is never active during the design; could be potentially removed.
[¶] = Pessimistic Design Constraints; constraints otherwise too competitive with those at the system-level.

The limited constraint scaling can be seen here in the order of magnitude variation between the spar buckling constraint and the rest of the listed constraints. The C_d on both the ribs and spar buckling show that these are the limiting factors within the bounded solution space, hence why they were the active constraints in the optima from the monolithic optimisers and the optimistic design cases. Within the normalised solution space, the rest of the constraints have very similar solution ranges.

When a broader design capability, C_d, is combined with a low design freedom, D_f, it is more likely that a given constraint will be pessimistic. Effectively, this situation represents a large solution space with a small range of feasible solutions. In the case of the upper and lower detailed stress, the original monolithic C_d for constraint responses are larger than the D_f within the substructure design problem.

6.5 Representative Wing Optimisation Problem

6.5.1 Design Problem

The design parameterisation of the GFEM and DFEM design problems allows for a variety of model and loading conditions. The following section describes a test of a representative wing design problem under static aeroelastic loads built and designed using the same optimisation framework as the baseline design problem. Figure 6.7 shows the wing, loosely based on the system-level interpretation of the NASA CRM wing, with a higher aspect ratio of 7.15. The half-wing GFEM has a span of 29.25 m and nominal chord of 4.091. It has a cross section based on the NACA 24020 profile, with varying sweep and taper along the span, and forward and rear spar based at 14% and 80% of chord, respectively.

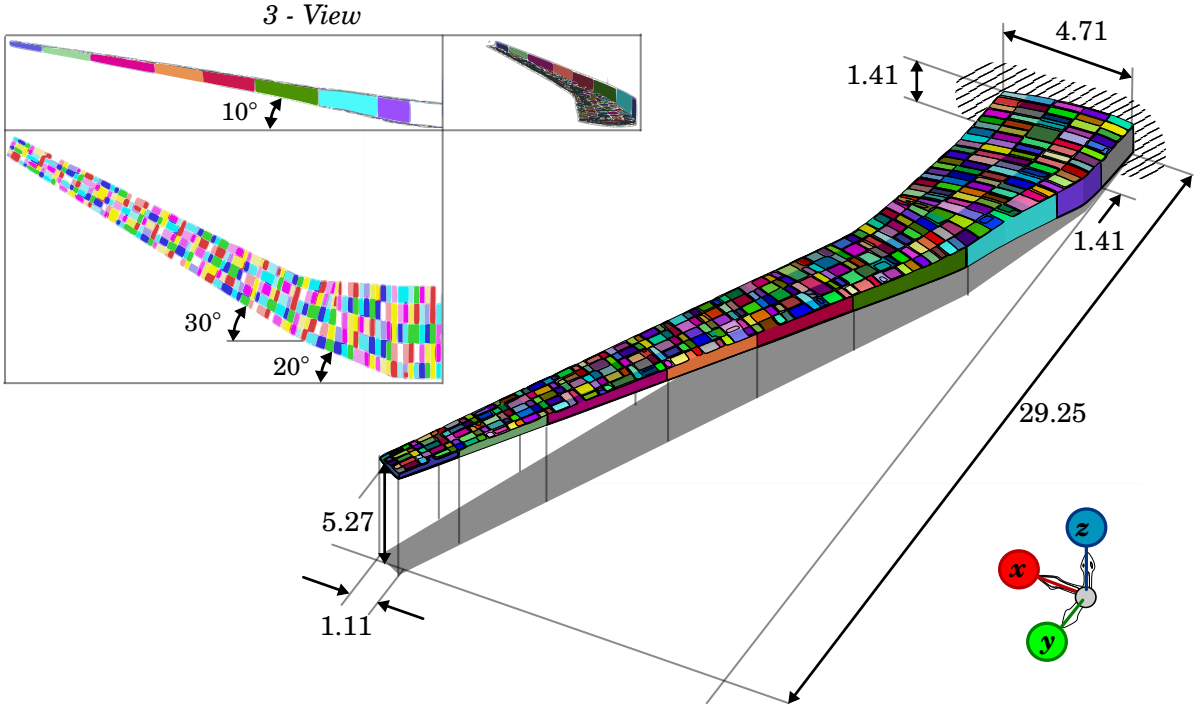


FIGURE 6.7. Representative Wing-Box System Level Model; FE Model and 3-View

The resulting wing-box features 27 internal ribs with varying orientations and variable rib pitch between 0.75m and 1.5m along the span of the wing. Skins are stiffened with T-shaped stringers with a constant pitch of 0.35 m and 0.7 m on the upper and lower skins, respectively. The wing-box metallic, the same isotropic 2000-series aluminium alloy used in the rectangular wing-box example. The system-level FE model has 4922 nodes and 5370 elements divided into 9 sections of variable size, resulting in $N = 18$ subproblems on the upper and lower surfaces.

The wing load distribution aims to be representative of a typical limiting scenario used in industry. Forces and loads were generated using an aeroelastic design case for a static angle of attack trim, Mach 0.85, and dynamic pressure 7,812 Pa (altitude ~ 43,000 ft) under a 2G gust lateral load. An additional 10 tonnes of a non-structural single point mass was at the root of the wing. The wing loads were taken from the GFEM at the initial design condition and fixed in order to simplify data transfer to subproblems.

There are 146 design variables disseminated as shown in table 6.6. The constraints use the same scaling as the rectangular wing-box model with 8400 constraints split between the system, and subsystem-level models; including the additional (*pessimistic*) stress constraints on the local panel models and full element coverage for system-level constraints. This model no longer uses sampling points along the wing for system-level constraints, instead it uses the information from all elements present in the model. This should alleivate any errenous sizing bias during the optimisation.

TABLE 6.6. Representative Wing-Model Design Variable Summary

Design Level	Description	Total
System Level, $\boldsymbol{x}$	Rib thickness	9
	Forward Spar thickness	9
	Rear Spar thickness	9
Linking Variables, $\boldsymbol{x}_0$	Upper Skin thickness	19
	Lower Skin thickness	19
Subsystem Level, $\boldsymbol{y}_{1-18}$	Upper Stiffener Dimensions	36
	Lower Stiffener Dimensions	32
	Manhole Plank Thickness	9

6.5.2 System Level Optimisation

The initial design was optimised using the system-level GFEM without additional subsystem constraints, figure 6.8 shows the iteration histories for this analyses. The best final solution converged to an optimised mass, $M^* = 10904.30$ kg, with a variation of 0.001% from multiple starting points in the design space.

(a) Single Design Point

(b) Multiple Start Objective Function Iteration History

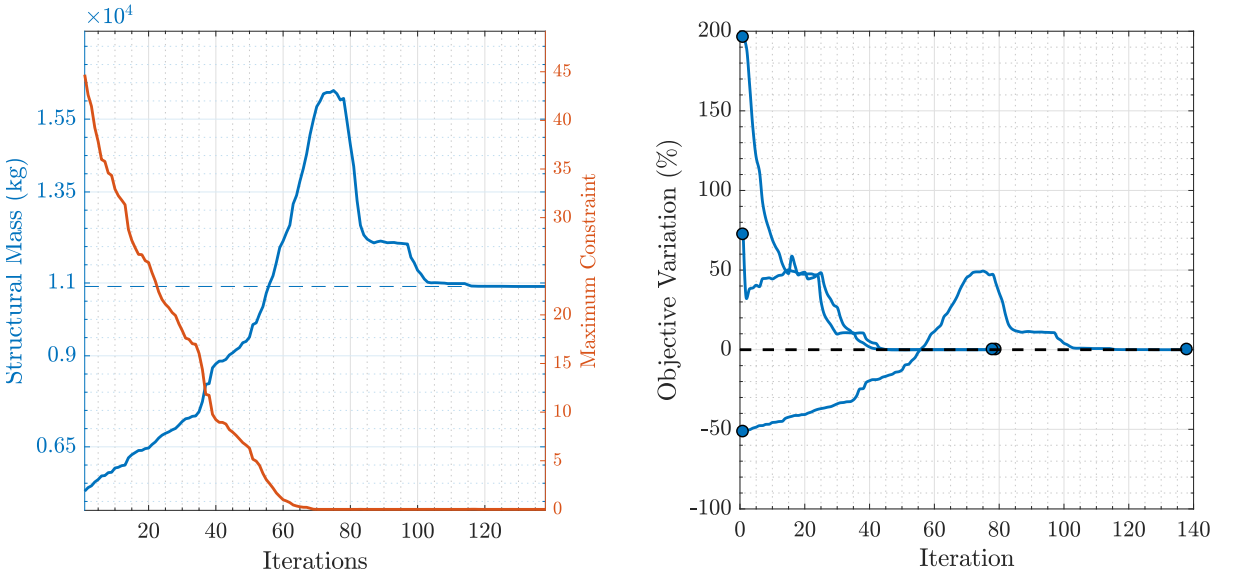


FIGURE 6.8. System Level Representative Wing Model Optimisation Iteration Histories; Satisfactory less than zero maximum active constraint and final optimised objective mass, $M^* = 10904.30$ kg±0.001%

The system-level results are consistent with the convex behaviour in the GFEM model based on the rectangular box-wing model. The constraints, substructure decomposition and response partition are based on the same criteria, so assumptions on I_C, D_f and C_d should remain consistent.

Figure 6.11(a) shows a design variable breakdown for the optimised system-level design. The model was optimised for system and linking variables $[x, x_0]$, the MHP thickness and stiffness dimensions were fixed at the mid-point between the upper and lower bound. The wing sweep of 20° starts in section 4, increasing to 30° by section 5. The corresponding sections on the wing feature increased skin thickness' in the ribs, spars and especially on the forward upper & lower wing surfaces.

The variation between the forward & after cover skin thickness for the upper and lower surface is consistent with the increased lateral twist in the swept portion of the wing. The increased lateral twist is the result of bend-twist coupling from a 2G gust load. The elevated rib

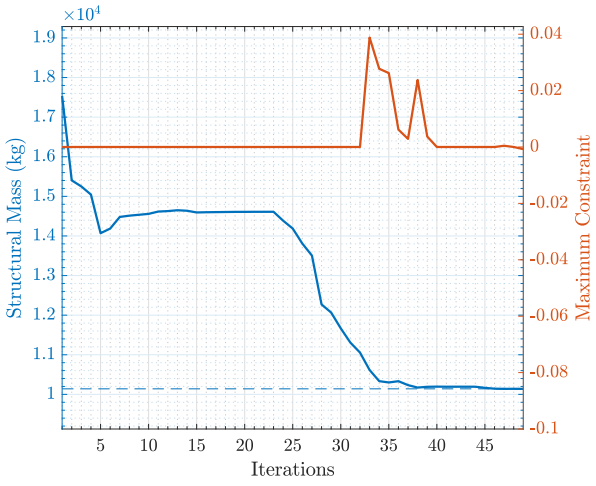
thickness over this area of the main body of the wing-box also supports this theory, with the rib thickness declining from section 5 of the wing on-wards.

Figure 6.11(b) is the constraint value distribution, for 7629 constraints taken from the system-level model. Substructure function responses were not included as part of this optimisation and were not included in the constraint distribution. The active constraints are a combination of the upper and lower skins in stress and the spars in buckling.

6.5.3 Monolithic: Single Phase AAO optimisation

Figure 6.9 shows the iteration history for the AAO optimisation applied to the representative wing model. Unlike the system-level design, the AAO function does not display convex behaviour across the majority of the design space; in order to find the optimum, the starting condition must be reasonably close to the final solution. The best solution found was $M^* = 10904.30$ kg, with a maximum variation of 34% from multiple starting points.

(a) Single Design Point



(b) Multiple Start Objective Function Iteration History

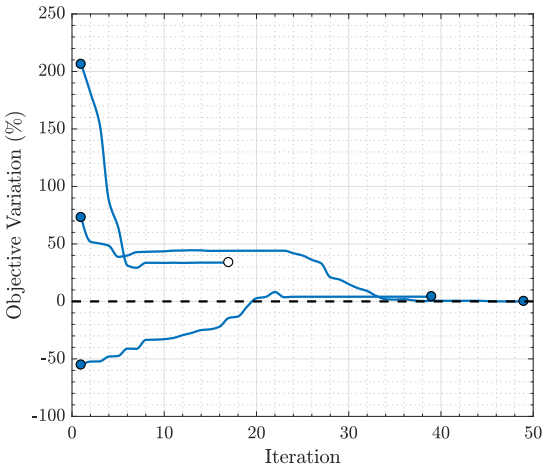


FIGURE 6.9. AAO Representative Wing Model Optimisation Iteration Histories; Satisfactory less than zero maximum active constraint and final optimised objective mass, $M^* = 10139.6$ kg ±34%

Figure 6.12(a) shows the design variable breakdown and a reduced constraint value distribution for the best AAO optimised design. The additional constraints have led to an increase in skin thickness sizing on both the upper and lower skin covers. The MHP and stiffener areas are variable, based on the changing size and shape of local panels. Including the additional models led to a mass saving of 7.01% compared to the GFEM-only optimisation in figure 6.11(a).

Figure 6.12(b) is a reduced constraint value distribution for the representative wing-box model, displaying only the top 10 most significant constraints per constraint set. It is more

evident that the active constraints for the AAO optima are a combination of the GFEM upper and lower skin stress, spar buckling, and DFEM MHP panel stress/buckling. The combination of system-level and subsystem-level constraints at the optimal design point indicate that both types of models were accurately presented in the optimisation.

6.5.4 Distributed: Augmented ATC optimisation

Figure 6.10 shows the outer-loop information for the dual-distributed gradient-based augmented ATC optimisation for the representative wing model. The analysis took 9 outer loops to reach model consistency between the distributed GFEM: DFEM models, with a final optimised mass $M^* = 10140.7$ kg, and MaxC of 0.0224.

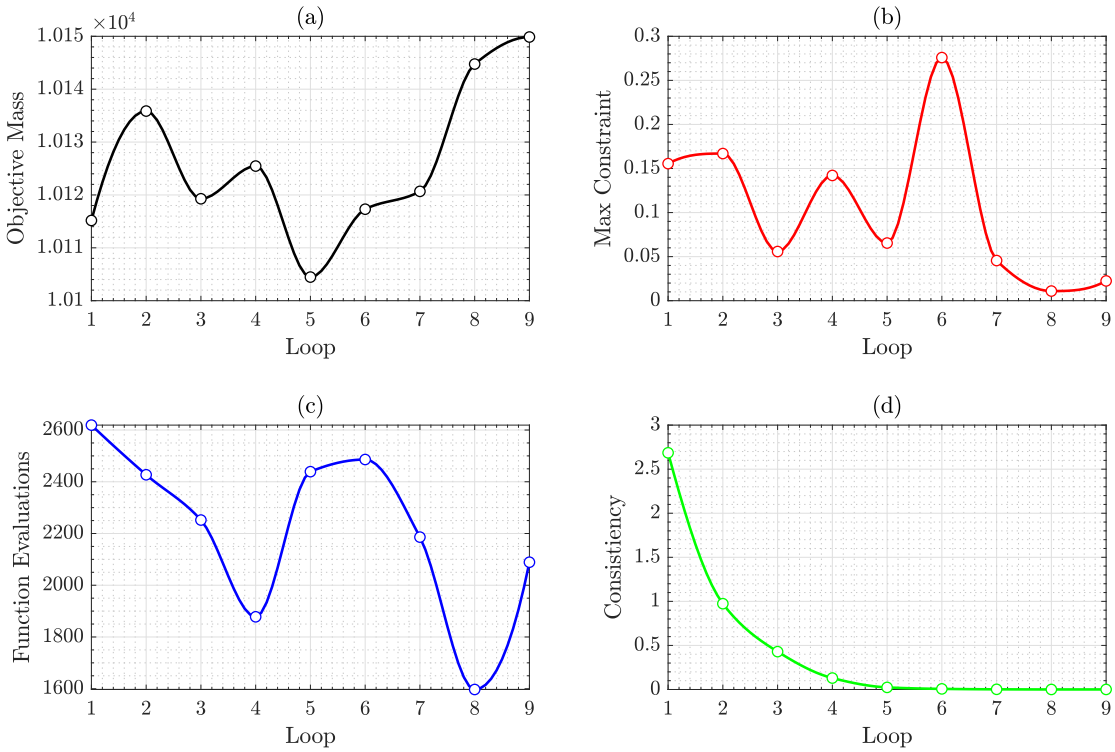
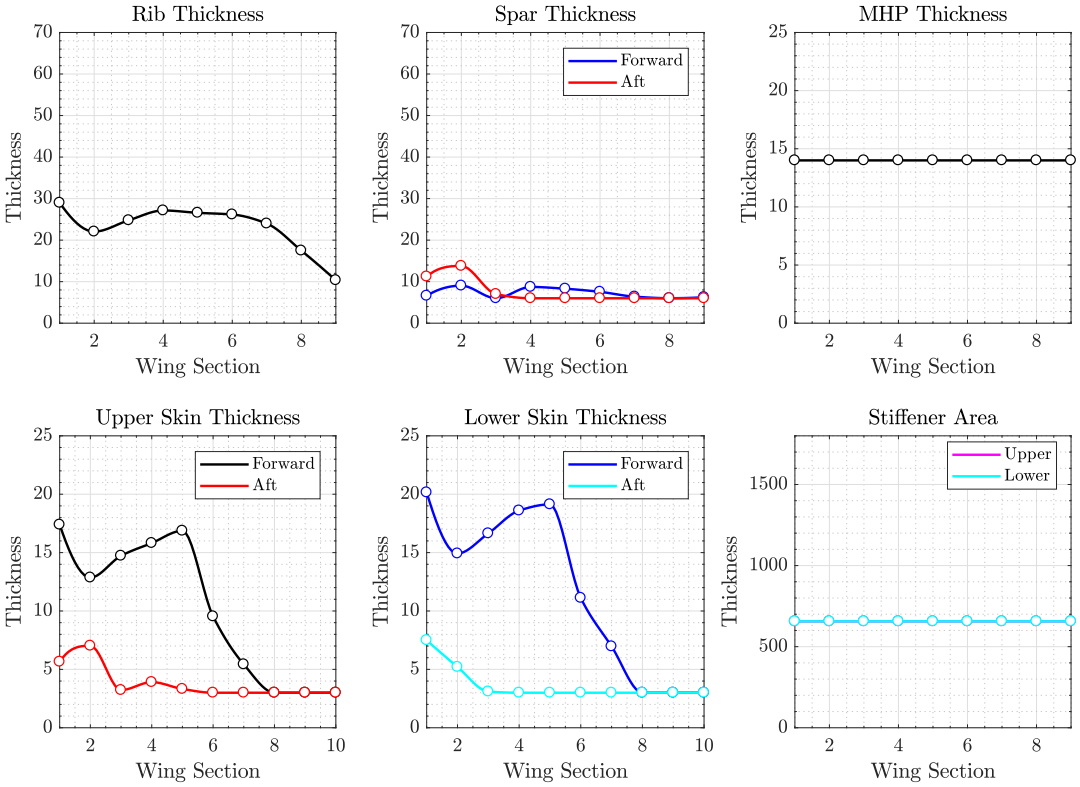


FIGURE 6.10. ATC: Representative Wing Outer Loop Optimisation;
(a) Objective Mass, (b) Maximum Active Constraint,
(c) Function Evaluations per Loop, (d) Model Consistency

Figure 6.13 shows the design breakdown and reduced constraint distribution for the final ATC design. The solutions for the AAO and ATC based designs are nearly identical, indicating both have converged to a potential global optimum for the given design problem. The violated constraints are from the spar lower panel MHP buckling.

(a) Design Variable Breakdown



(b) Constraint Value Distribution (total: 7629)

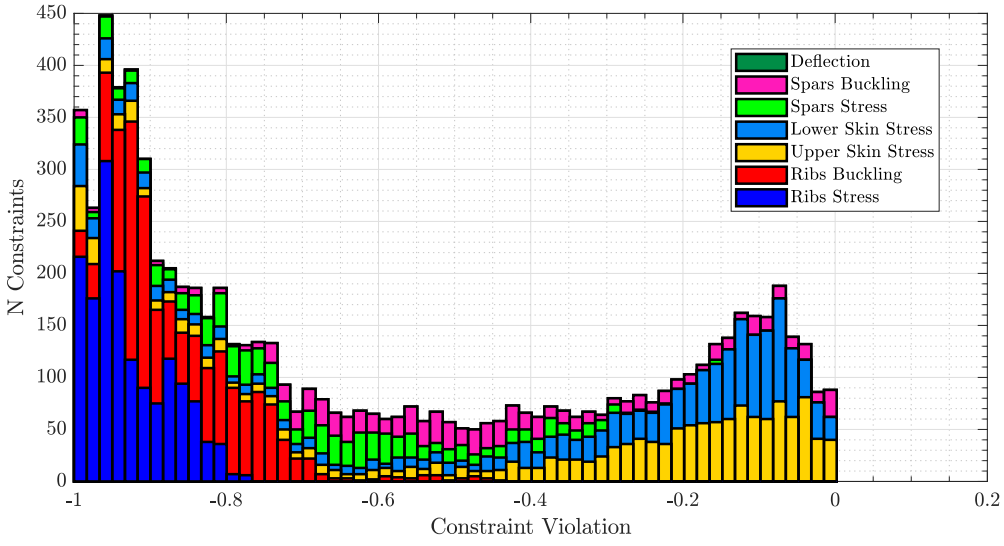
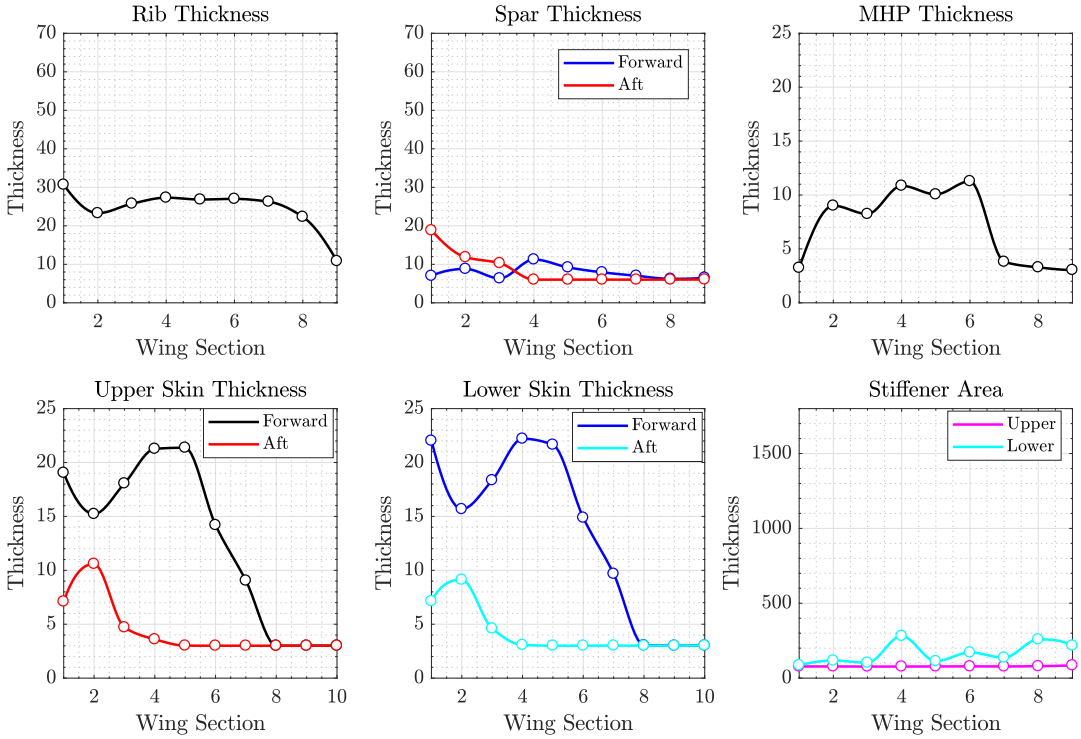


FIGURE 6.11. System Level: Representative Wing-Model Final Design Summary; Stiffener Dimensions & MHP Design at initial Conditions, $M^* = 10904.3$ kg

(a) Design Variable Breakdown



(b) Reduced Constraint Value Distribution (total: 8400)

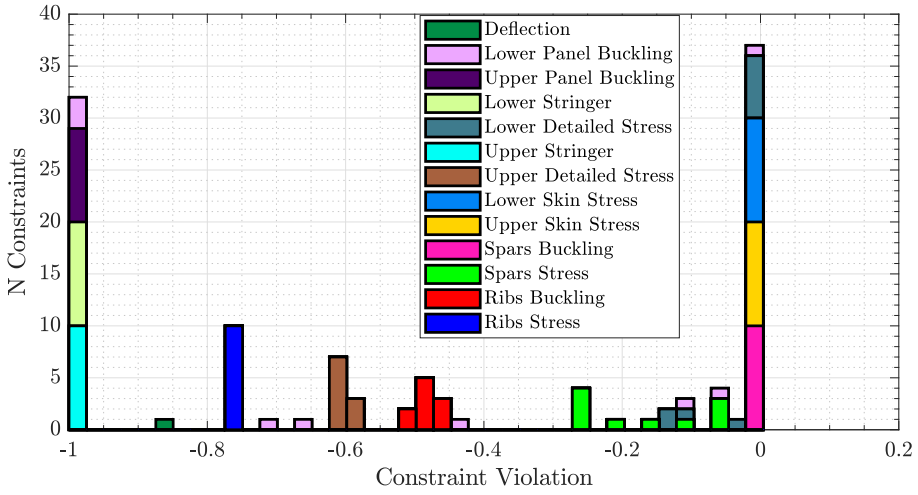
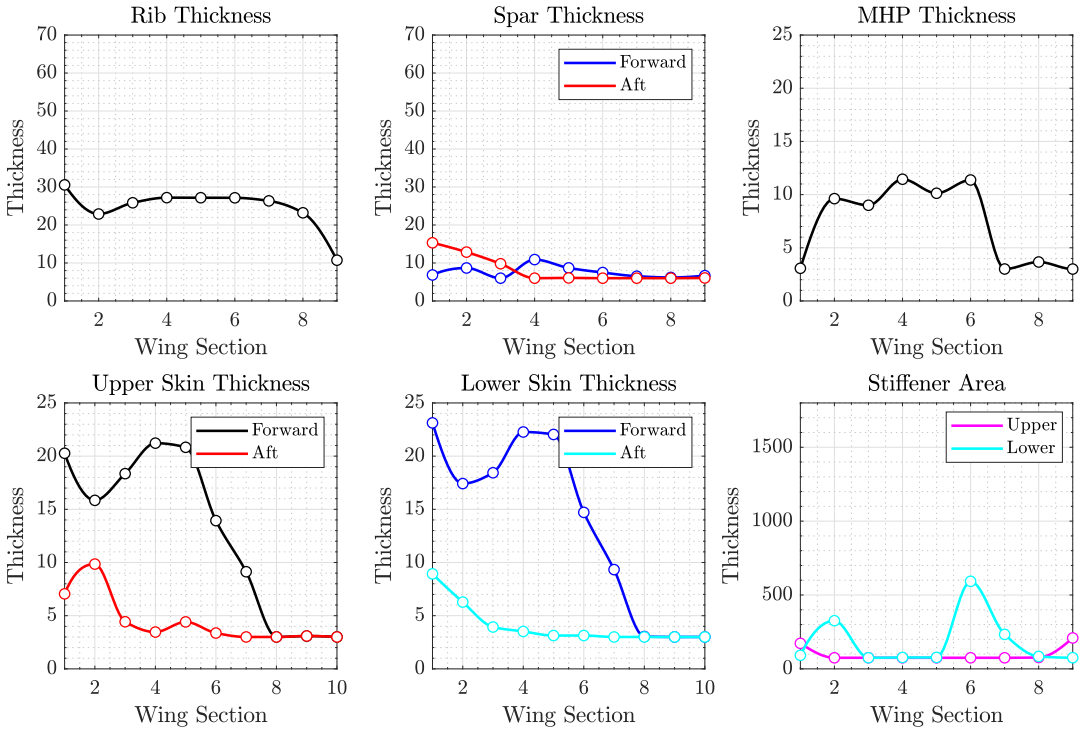


FIGURE 6.12. AAO: Representative Wing-Model Final Design Summary; Constraints displayed reduced to top 10 values per constraint, $M^* = 10139.6$ kg

(a) Design Variable Breakdown



(b) Reduced Constraint Value Distribution (total: 8400)

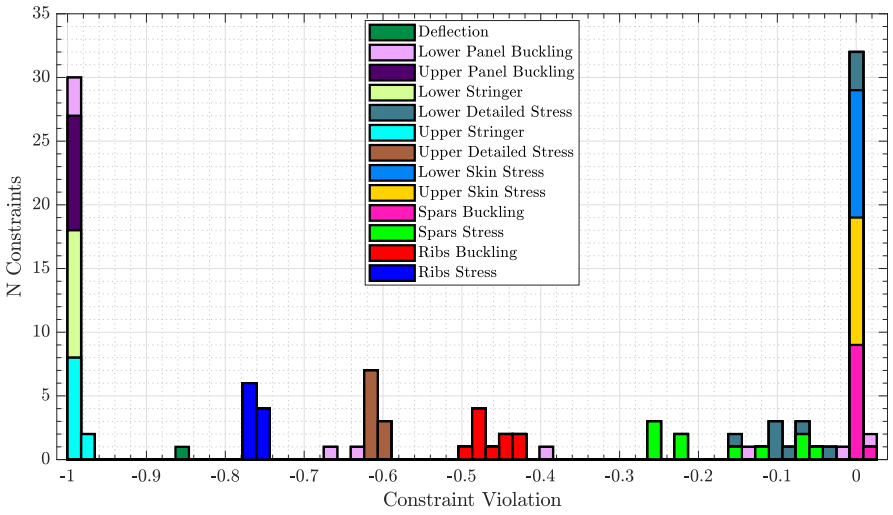


FIGURE 6.13. ATC: Representative Wing-Model Final Design Summary; Constraints displayed reduced to top 10 values per constraint, $M^* = 10140.7$ kg

6.5.5 Results Summary

Figure 6.14 shows the objective iteration histories for the AAO and ATC methods with respect to the GFEM-only design optima. Table 6.7 summarises the optimisation results for each method, and features the best & worst AAO design results.

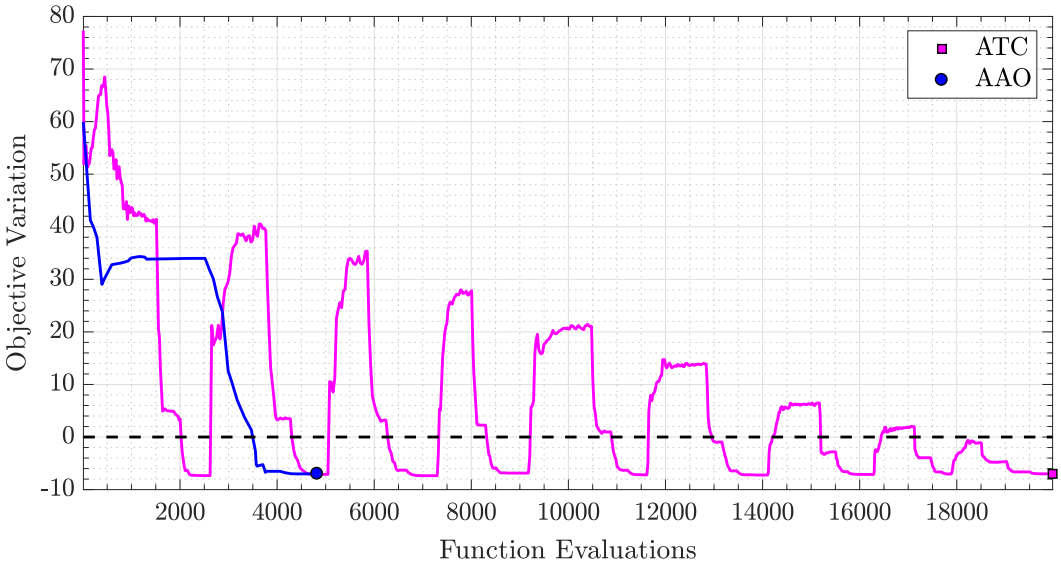


FIGURE 6.14. Results for the Wing Model Optimisation Problem; Function evaluations vs. Mass Variation from MACRO optimised solution, M^* = 10904.3 kg.

	evals	M^*		MaxC	nCon	sum(v^*)
GFEM-only	256	10904.3		0.000 (0.217)†	7629	51.156
AAO^1	2033	13561.2	+24.37%	1.658	8400	92.873
AAO^2	5396	10139.6	-7.01%	-0.001	8400	19.939
ATC	19973	10140.7	-7.00%	0.022	8400	20.921

^{1} = worst AAO design, ^{2} = best AAO design, † = GFEM-only design + DFEM constraints

TABLE 6.7. Summary of Optimisation results for Representative Wing-Model

For the wing problem, the AAO and ATC methods converge to near-identical designs; both have additional mass savings compared to the GFEM-only optimisation. The monolithic AAO solution was dependent on the starting position, in the worst case converging to an infeasible solution. When given an initial position reasonably close to the starting position, AAO converges to a solution identical to the more robust ATC method. The ATC method required ~ 3.7 times the function evaluations of the AAO method.

For the presented design problem, it would be more efficient to compute multiple AAO solutions, updating the starting position of each to reduce the difference between the initial

position and optima. However, as shown with the rectangular wing model, monolithic methods may not be robust enough to apply to the most challenging design problems.

The GFEM-only solution when run with the complete set of constraints has a maxC of 0.217. Using this solution as a starting-point for the next optimisation was tested but without using multiple starting positions as in the SEQ method, it converges to an unfeasible solution.

Figure 6.15 shows the skin cover distribution for the upper and lower surfaces at the optimum solution the GFEM-only, AAO and ATC methods. The similarities between the final AAO and ATC designs is more evident when directly comparing the skin covers. Both designs feature peak thickness' for the upper and lower surfaces at the root and near the kink in the wings planform, while the value distribution is near identical. The improved final solution using the AAO method may be due to increasing the number of constraints to encompass as much of the system-level and subsystem-level behaviours as possible.

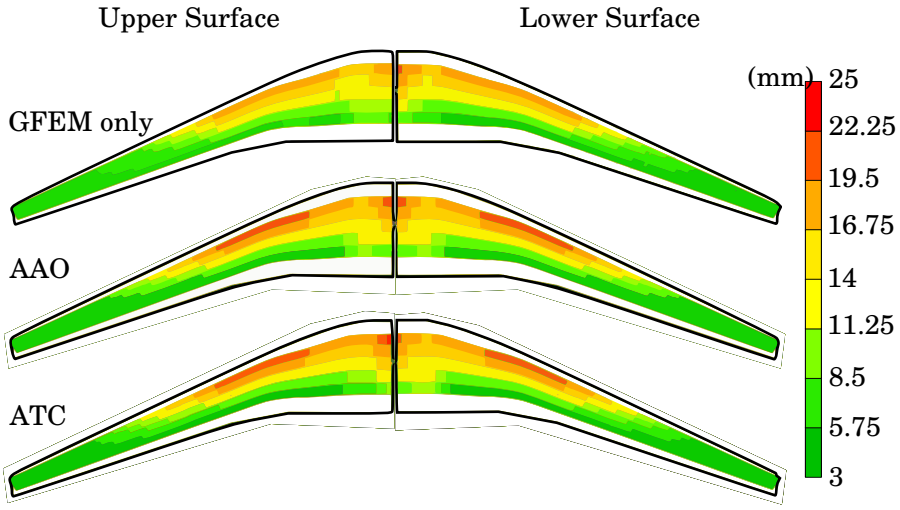


FIGURE 6.15. Optimised Thickness Distributions for Upper and Lower Surfaces; Comparison of GFEM-Only, AAO and ATC solutions

The current industrial approach would be to optimised for the system-level only model at the preliminary design stage, and later include the detailed panel and substructure constraints. The design represented by the GFEM only model would be the result of this approach. Both the AAO and ATC methods include a greater amount of detail within there design and the increased refinement has lead to changes in the overall skin cover sizing, seen in figure 6.15. Additional models would be likely to increase the deviation between these models; these include but are not limited to: Spar and rib topology optimisation, engine mountings, fuselage interfaces or control surface fixtures.

6.5.6 Problem Profile Summary

The rectangular box-wing model featured two different design assumptions; one optimistic with a favourable and more straightforward to solve local subproblem, and the other pessimistic with a detrimental and more challenging to solve local subproblem. The representative wing model problem features a combination of optimistic and pessimistic subproblems, a more likely scenario with a real-world design problem.

Figure 6.16 shows the constraint PDF for the upper and lower panels, with the feasible and infeasible design spaces highlighted in green and red, respectively. The upper panels, figure 6.16(a), features optimistic design subproblems. On the upper panel, each PDF shows complete design freedom, with the design capability narrowing as the constraint responsiveness to design variables, reduces along the wing. Figure 6.16(b) has a selection of pessimistic responses inboard of the wing, section 1 to 6. The lower panel outboard response, sections 7 to 9 are more optimistic, indicating the stress-buckling constraint problem become simpler to solve.

TABLE 6.8. Representative Wing Model Decomposition Profile

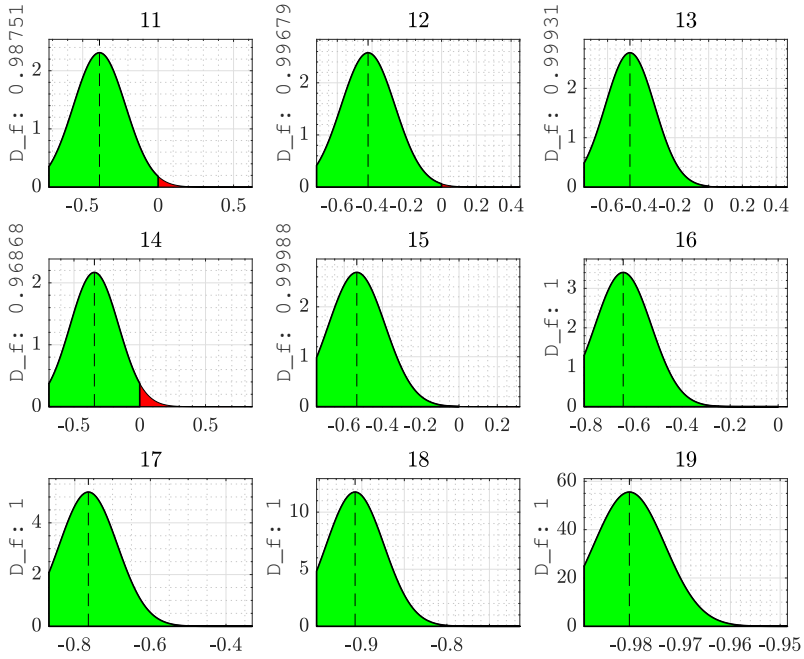
	Objective I_C	$\vec{D_f}$	$\bar{\sigma}$	$\bar{\mu}$	$\vec{C_d}$
Upper Panels (1 - 6)	1.000	0.992	0.154	-0.467	0.028
Upper Panels (7 - 9)	1.000	1.000	0.039	-0.884	0.002
Lower Panels (1 - 6)	1.000	0.268	0.435	0.195	0.174
Lower Panels (7 - 9)	1.000	0.989	0.123	-0.696	0.018

Table 6.8 measures the inboard-outboard solution space shift from optimistic to pessimistic. The gradient-based decomposition method leads to a meaningfully decomposed set of subproblems across the entire wing with $I_C = 1$. The upper panel D_f broadens as subproblems move outboard, while the C_d narrows as the range of possible solutions reduces.

The root-most lower panels (1-6) have the lowest design freedom with the highest design capability. As stated previously this means the majority of the design space in comparison to the other subproblems is infeasible, with the most failure states for any given solution. This means it is likely there is a pessimistic relationship between the system-level and these specific subproblems.

The result is an entirely optimistic set of upper-panel subproblems and a combination of optimistic/pessimistic lower-panel subproblems. Note, that even when there is a handful of pessimistic models present it can lead to convergence difficulties when using monolithic methods.

(a) Upper Panels; 11 root ⟶ 19 tip



(b) Lower Panels; 21 root ⟶ 29 tip

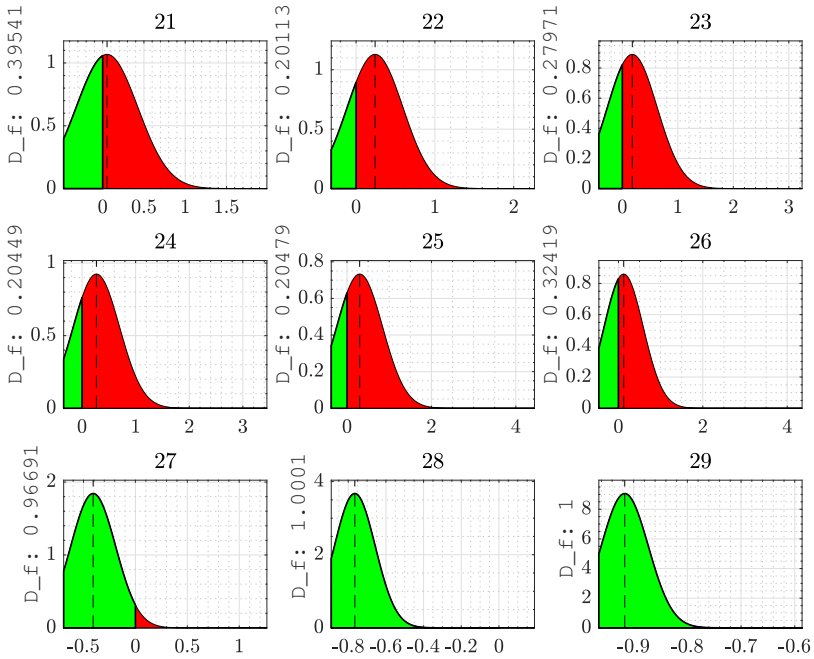


FIGURE 6.16. Representative Wing Model Local Problem PDFs; the feasible design space and infeasible design space of the upper and lower subproblems at Optima.

6.6 Chapter Summary

This chapter presented a comparison between the significant subsets of the optimisation literature, with additional quantitative analysis of the applicability of each method to the problems presented. Salient information presented herein includes a comparison of monolithic and distributed methods, how design assumptions support the convergence criteria of the algorithm as well as model and problem profiling indices. A definition of meaningful decomposition and how information certainty measures can be used to ensure the appropriate partition of design variables and objective: constraint responses.

The main design considerations highlighted with the MLO architectures were:

- *Monolithic vs Distributed methods:* Selection criteria and the application of each;
  - Monolithic solvers are less robust and after multiple attempts may not get anywhere close to the best solution. However, they are orders of magnitude more efficient than distributed methods for large-scale design problems.
  - Distributed methods are very reliable when tuned correctly. However, the breadth of architectures, state-parameters and decomposition considerations means it can be challenging to select appropriate methods, set-up and produce meaningful results.
- *Meaningful Decomposition:* Any partitioned design problem must capture a sufficient portion of the primal problem response. Information certainty, I_C, can be used to assess how much of the solution space is available to a partitioned subproblem. The lower the certainty, the narrower the solution space, the more likely to have fundamentally change the design problem, resulting in design subproblems which cannot be solved, or inefficiencies in the outer loop coordination of the overall problem.
- The less drastic the difference between the system and subsystem level modelling, the more stable the local problem performance range is to external variables. The result for the designer is that the more design information idealised in the system-level model, the simpler the MLO problem is to solve.
- *Pessimism in subproblem Relationships:* Pessimism in design engineering was more robustly defined, identifying the range of effects on the MLO problem. Pessimism depends on a range of conditions, including but not limited to:
  - When the subproblem constraints were more adverse and more likely to be active competing constraints at optima.
  - If there were significant differences between optimised solutions at the system and subsystem level due to local constraints or differences in physical behaviours or geometry

  - If there is significant variance in local subproblem function responses due to system-level or adjacent subproblems.

- Design freedom (D_f), and design capability (C_d) can be used to identify pessimism within uncertain designs. Lower design freedom, combined with a higher design capability indicates adverse local constraints present increasing the difficulty of the optimisation problem and the likelihood of pessimistic problem relationships.

- *Solution Space Constrain Screening:* The partition of variables and responses into subproblems mist consider the scaling, convexity and the type of data combined. Ideally, grouping distributed responses with similar optima and design capability simplify the MLO solution.

Design decomposition and problem partitioning can create unique non-repeatable problems, making it difficult to benchmark analyses or apply learning directly to an industrial environment. It is possible either by constraint selection or model partition to fundamentally change the presented design problem. The result is masking the complexity of individual models and multi-model relationships in the process of trying to solve for the multi-model problem.

The best multilevel methods can account for nuances in the MLO problem by including additional or iterative out of loop optimisation. However, in an industrial setting the best approach would be to create a multilevel model within a flexible framework. As seen in the literature there are a breadth of successful methods but, as discussed, minor problems changes and design assumptions can change the viability of one method over another.

Any robust MLO framework must allow for several different methods to be tested with an acceptable monolithic baseline. The possibility of multiple optimisers, numbers and levels of subproblems and integrating selection and quality assurance criteria for presented design problems. Within an industrial optimisation environment standardisation of inputs, constraint and design scaling with common non-black box interfaces could be used to integrate multi-model, and cross-team design challenges.

CHAPTER 7

FINAL CONCLUSIONS & FUTURE WORK

7.1 Final Conclusions

This thesis has presented an investigation into multilevel structural design optimisation for aerospace structures, starting with a review of the existing work within the literature in chapter 1. Through the literature a broad range of methods and sizeable number of applications were found for MLO and cooperative design models. The breadth of research was compiled into several illustrative descriptions of algorithms, grouped under monolithic, dual and primal distributed architectures.

Chapter 2 described the development of parametric models for system-level wing structural design and related substructures—with preliminary investigations into optimisation of the resulting problems. Chapters 3, 4 and 5 integrated parameterised models into MLO architectures representative of each of the major areas with the literature. Each optimiser was assessed in-turn for an optimistic and pessimistic design case, featuring a favourable and adverse subproblem relationship, respectively.

The results of each optimisation were collated in chapter 6 comparing optimisation results in terms of efficiency and quality of final solution — a range of quantitative metrics were identified for the applicability of each method to a given design problem. The most successful and efficient algorithms were then tested on a representative wing-box model under aeroelastic loads.

The main outcomes of this research can be summarised in the following points:

1. This work provided a review of the scope and sizeable applications of MLO in hierarchical engineering design and the broader research area. This review identified the salient portions of decomposition theory and multilevel algorithms, with applications of MLO in engineering design, subdivided into the primary architecture types: monolithic, primal and dual distribution.

2. For practical use, the goal of an optimiser within engineering is to produce a potential solution to a design problem, that is, an improved solution but not necessarily the exact and best optimum value. In this case, single-phase monolithic optimisers are the most efficient,

requiring just 20-25% of the number of function evaluations compared to distributed and multi-phase functions. However, under a range of assumptions a meaningfully distributed problem using gradient-based augmented ATC leads to a 4-7% mass reduction over the monolithic AAO optimised solution.

3. The relationship between the system and subsystem level problems can significantly change the convergence properties of monolithic solvers – independent of the convexity of any individual function. Pessimism is when a competitive design relationship reduces the feasible solution space overlapping in multi-model design problems. Monolithic solvers can produce erroneous and overall inadequate design solutions under pessimistic conditions. Pessimism depends on a range of quantifiable conditions:

   – Whether the subproblem constraints are more adverse and more likely to be active constraints at optima.
   – If there were large differences between optimised solutions at the system and subsystem level due to local constraints or different physical behaviours and geometry.
   – If there is significant variance in local subproblem function responses due to system-level or adjacent subproblems.

4. This work identifies a range of problem metrics used to quantify the quality and necessity of optimisation by partition and decomposition, as well as sources of error within MLO design problems. Information certainty (I_C) can be used to assess how much of the solution space is available to a partitioned subproblem. The lower the certainty, the narrower the solution space, possibly changing the overall MLO design problem and drastically reducing optimisation efficiency. Design freedom (D_f), and design capability (C_d) can be used to identify pessimism within uncertain designs. A subproblem with narrower a design freedom combined with broader design capability increases the adversity of local constraints present, the hardness of the local problem, and the likelihood of pessimistic problem relationships.

5. For a representative wing-box model under aeroelastic loads both monolithic and distributed MLO exploited additional design margins compared to a single level optimisation model using conservative estimates. This approach was based on best practise used in the current industrial preliminary design environment. The additional refinement at the preliminary design stage allowed for a more exhaustive solution space and additional weight saving while maintaining feasibility under additional local constraints.

6. Problem partitioning and design decomposition can create unique non-repeatable problems. Ultimately making algorithms difficult to benchmark analyses. Depending on how MLO is implemented, a designer could fundamentally change the range of feasible solutions when compared to the non-parsed primal for of the problem. The best multilevel methods

maintain the nuance of the original problem by including additional coupling variables or iterative out of loop parameters.

7. As part of this study the marked difference between

8. Concluding recommendations for the best approach for MLO within an industrial environment would be to create a flexible framework with the following properties:

   – Modular in scope and capable of general expansion with standardised of inputs, constraint and design scaling with standard interfaces.
   – Standardisation of models is crucial to allow for different corroborative optimisers.
   – Flexible enough to allow variable constraint selection, assumptions and types of partition.

7.2 Future Work

Additional research is required in the following areas:

1. Assess the viability of quasi-separable assumptions with potentially shared subproblem objectives and constraints — additional considerations including volume, fuel weight and part manufacture, accommodation or inspection.

2. Additional testing in the use of different types of optimisation algorithms within single-phase monolithic architectures. This research focused on the SQP methods present within MATLAB, but additional gradient-based algorithms are available. Potential methods include the Globally Convergent Method of Moving Asymptotes (GCMMA [151]) and the coupled conjugate-adjoint method [21, 22].

3. Testing how the efficiency of monolithic vs distributed methods varies with the number of variables, responses and subproblems. Increasing the number of subproblems should scale functions evaluations quadratically using monolithic architectures, while distributed methods should scale linearly. Theoretically, the higher the number of subproblems within a design, the less indomitable monolithic methods become. However, it is uncertain whether the increased number of subproblems negatively impacts the upper-level coordination problem.

4. One of the benefits of MLO is introducing detailed localised designs earlier into the design process. By introducing levels of design above the 3D half-wing GFEM, it would be possible to integrate design considerations at the "prototype" full-wing or full-aircraft level. Including design requirements for fuel and engine accommodation, additional flight and loading conditions.

5. Non-generic substructures with additional design features and function responses were found to increase the uncertainty in monolithic design solutions. Introducing more detailed design levels beyond the panel level DFEM used in this research explores the issues of scaling within the design space. A "micro" design level would include material, manufacturing or laminate properties within a composite based wing design. This domain expansion would widen the scope of applicability of methods within the modern engineering design environment.

6. The system-level and subsystem level design problems both used NASTRAN as the primary solver. Additional investigations could focus on the validity of using computationally expensive detailed substructure models in comparison to analytical or conservative design guidelines for local components. Ideally, the higher the accuracy of elaborate and expensive models increases the confidence in the final solution. However, lower cost analytical methods may allow for the use of robust, globally optimal stochastic algorithms, potentially improving the final design overall.

7. MLO could be especially useful within the multidisciplinary design environment, where an N^2 formulation may result in a limited representation of critical systems.[1] A more realistic design environment would include the vertical integration of detailed design considerations into the MDO conceptual design phase. The MDO design problem would be expressed as a N^3 system model, see figure 7.1. Additional problem complexity would come from linking variable handling, the rationality of quasi-separable assumptions and interfaces for cross-discipline requirements, e.g. engine pylon structural design → Propulsion cowling aerodynamics.

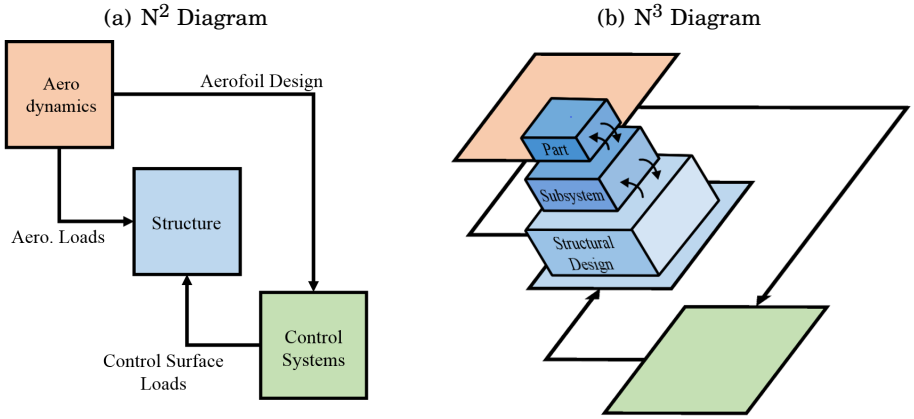


FIGURE 7.1. Multidisciplinary layout with Multilevel System elements

[1] An N^2 diagram is a matrix, representing functional or physical interfaces between system elements.

8. Finally, expanding the parametric model used to include non-continuous design variables and features. Additional features may include material selection, composite layup, number and position of internal components and more. The best monolithic solvers currently cannot adequately handle the integration of mixed-integer or discrete combinatorial design problems. These are design problems for which distributed methods are a necessity by nature. A new comparative study could investigate parametric modelling methods between using representative continuous design variables, such as percentage of $0, \pm 45, 90^\circ$ laminates, with a discrete combinatorial subproblem optimisation for ply layup.

“ *Things didn't go exactly as planned, but I'm not dead, so it's a win.* ”

Andy Weir, *The Martian*, 2011

APPENDIX A: PARAMETRIC WING MODEL

This section details the capabilities and model idealisation of the parametric model generation used for the system-level models.

A.1 Parametric Wing Design

The design automation process is based around a representative MATLAB® wing model and data-set. Accurate model decomposition requires exact data transfer of mesh data to reoccurring substructures. The only way to ensure consistency with subsystem models is to fully parameterise the structural model, where data can be stored in a repository and modified, duplicated or accessed by various models.

The systems layout shown in figure A.1 is modular, based on the model-view-controller architecture [152], with independent object-based programmes representing the major modelling components[1]. Data processing and model evaluation have been separated, allowing each to be modified individually into the components identified in figure A.1. New FE models, processing methods or optimisers can be added without changing the model inputs or outputs between each module. Local copies of model data can be made for N subsystems, ensuring models can occur independent and concurrently with one another. The 'FE Wrapper' and 'Optimiser Data Object' are the same regardless of the FE model used, ensuring consistent data inputs and output screening [153].

[1] Development strategy where components in an application are developed based on decoupled layout, with independent components for concurrent development.

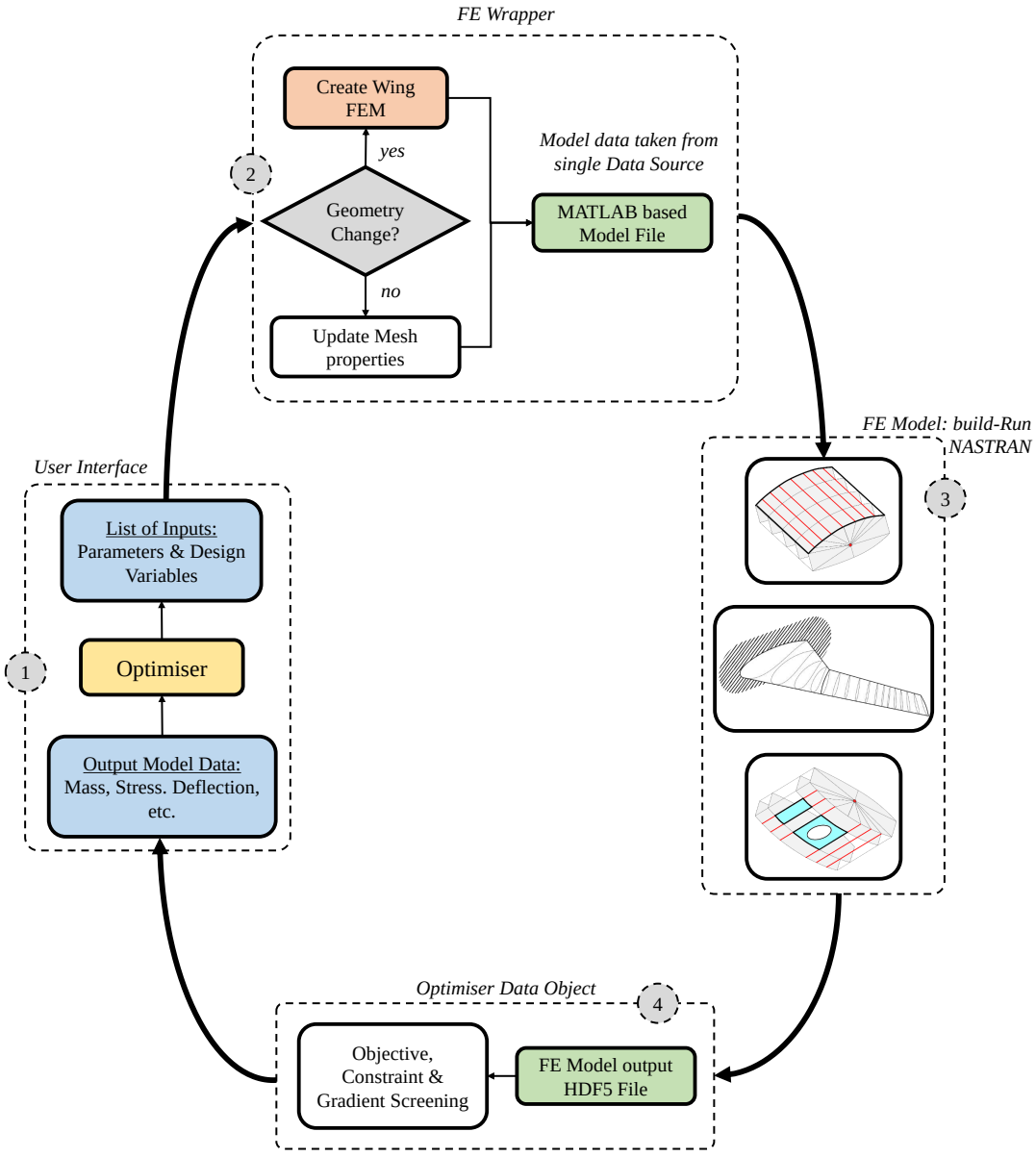


FIGURE A.1. Structural Systems Layout, represents the generic parameterisation used for parametric model generation, steps 1 to 4

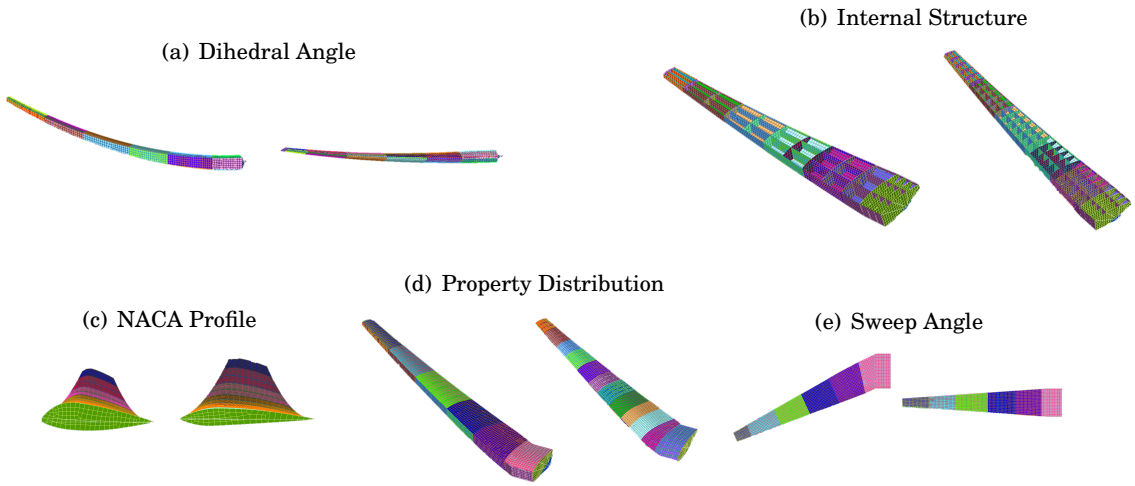


FIGURE A.2. Parametric Wing Model Variation

TABLE A.1. Parametric Wing Model Design Quantities

FE Mesh Data	Wing Input Data	Skin Thickness Inputs
Mesh Coordinates	Wing Profile	Upper Skin
Property Distribution	Span	Lower Skin
Spar-Rib Interface nodes	Chord Variation	Ribs
Rib-Skin Interface nodes	Twist Variation	Forward Spar
	Dihedral Variation	Rear Spar
	Number of Ribs	Stiffener Shape
	Spar Positions	
	Matireal Definition	
	Skin Stiffener Definition	

A range of parametric methods exist for whole wing, and full aircraft geometry mesh generation [128, 154]. These were considered for this work, but the requirements to decouple model building, objective and constraint screening, within a parallel-capable layout for 3D meshes, meant that a bespoke method had to be created. This layout is aimed specifically at concurrent 3D mesh generation with the aim to replicate the levels of detail available within industry. The inputs and data stored in the model have been compiled in table A.1.

It is possible to generate and store the model data for a range of external wing geometries, number of internal structures and property resolution for a 3D half-wing FEM. Figure A.2 shows the range of GFEM wing designs that can be produced by this parametrisation. The data generated is stored with loading, geometry, boundary conditions and mesh data structured to ensure consistency across multiple models and sub-models.

A.2 Model Idealisation

Ribs and spars are idealised as 2D plate elements within NASTRAN. The half-wing root constraint is a pinned joint about the rib, spar and skin interfaces with the root-most panel. Skin cover sizing is parameterised using $2N$ equally spaced control points along the forward and aft edges (N points per edge), resulting in a span and chord-wise thickness variation for each sub-structure, plus an additional point at the tip ($2N+1$ control points in total). The thickness follows 2D Lagrange distribution between adjacent control points, with an exemplar distribution shown in figure A.3 [130].

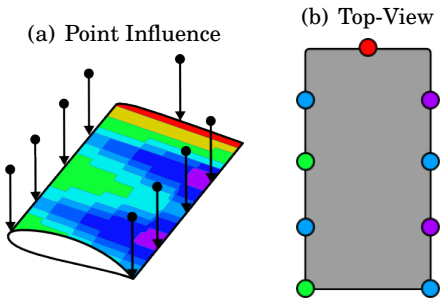


FIGURE A.3. Cover Sizing Control Points; Skin thickness distribution example for a short box-wing with number of section, $N = 4$.

Stiffener modelling at the system-level uses an equivalent composite laminate to represent the smeared mass and stiffness properties of a unidirectional stiffened panel. The properties are calculated by solving for the equivalent skin thickness, density and material properties of an additional layer laminated onto the skin as shown in figure A.4. The contribution of the bending stiffness for each component on the stringer can be evaluated exactly. Density and neutral axis position can equated to those of an equivalent laminate panel [131] with properties defined in equations (A.1), (A.2), (A.3), and (A.4).

$$(D_{11})_{total} = (D_{11})_{equiv} \tag{A.1}$$

$$(A_{11})_{total} = (A_{11})_{equiv} \tag{A.2}$$

$$(GJ_{11})_{total} = (GJ_{11})_{equiv} \tag{A.3}$$

$$(\rho A)_{total} = (\rho A_{11})_{equiv} \tag{A.4}$$

Here $(D_{11})_{equiv}$,$(A_{11})_{equiv}$,$(GJ_{11})_{equiv}$,$(\rho A_{11})_{equiv}$ are, respectively, the bending, axial and torsional stiffness, and the total panel mass of the equivalent model. Assuming the neutral axis position remains fixed, equations (A.1), (A.2), (A.3), and (A.4) result in a set of non-linear equations to be solved to produces an artificial laminate of equivalent mass and stiffness. The underlying skin thickness, t_{skin}, is unchanged.

The NASTRAN-based sensitivity analysis used herein for the stiffened panel optimisations that ensue is limited in scope, and cannot output GFEM behavioural gradients with respect to the resulting artificial laminate properties. Local stiffener dimensions have been assumed to act as local variables to the subsystem analysis.

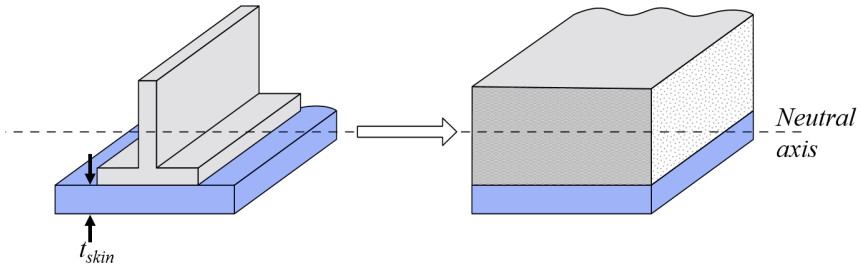


FIGURE A.4. Equivalent Laminate Panel; Assumes the neutral axis position remains fixed, solving for an equivalent laminate, with properties representing the smeared mass and stiffness properties of the original panel.

The material and geometric properties of the equivalent panel are applied to the wing upper and lower skins as composite laminates, allowing for a low order modelling of the stiffened panel behaviour. Using equivalent laminate-based stiffened panels also makes automated meshing and data transfer simpler to substructures, reducing the need for more sophisticated modelling at the systems level.

APPENDIX B

APPENDIX B: SUBSTRUCTURE MODELLING

Any substructure model generated must accurately represent the external forces acting upon it, whilst providing sufficient additional resolution and data to justify its partition. The following section outlines the investigation into a selection of sub-structuring methods ensuring subsystem models used the most appropriate methods available.

B.1 Super-element Modelling

There are four primary methods of super-element (SE) modelling within NASTRAN. Each type defines the SE model and transfers data from the SE to the residual model differently. The implementation of each was assessed for accuracy, and gradient extraction on a baseline model, these methods are:

- Part SE,
- List SE,
- External .pch SE (2-Step Method)
- External .xdb SE (3-step Method)

The baseline test used the GFEM with 1D beam stiffener representation, replacing composite laminates, and formatted with a fully fixed root constraint. The upper skin covers at the root, mid-span and tip most panels were assessed without any variation in model fidelity. The baseline buckling modes and corresponding eigenvalues are shown in figure B.1. Mode 1 sees the mass in the root panel reacting, Mode 15 is the first time the mid-span model reacts, whilst the tip most panel does not react within the first 40 modes.

(a) Mode 1 (0.458) (b) Mode 15 (0.708) (c) Mode 40 (1.013)

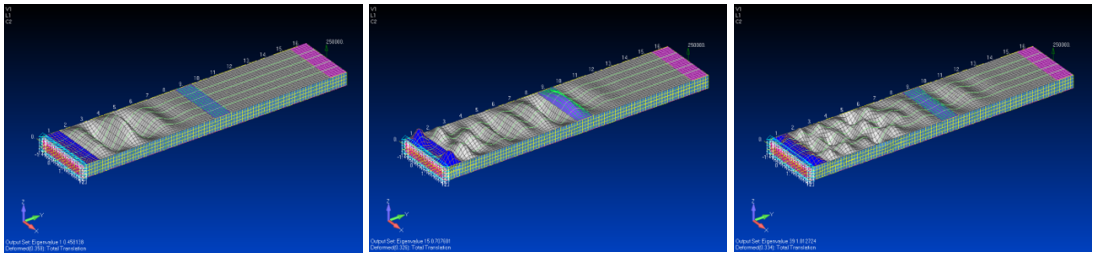


FIGURE B.1. Baseline Model; Scaled buckling mode shapes (Eigenvalue).

The local buckling shapes for the upper skin cover at the root, mid-span and tip, are compared in figure B.2, shown using the results from the LIST SE analysis.

(a) Root 1st Mode (0.653) (b) Mid-span 1st Mode (0.925) (c) Tip 1st Mode (14.262)

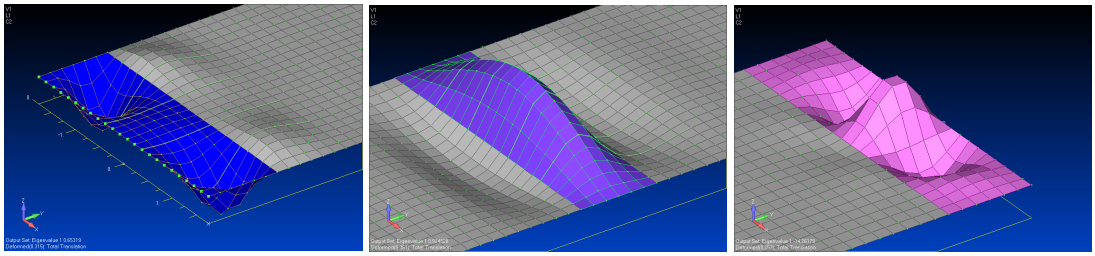


FIGURE B.2. Upper Skin Cover Panel Models - Scaled buckling mode shapes (Eigenvalue).

The root and mid-span eigen-values are the same order of magnitude as the global behaviour exhibited in figure B.1, such local interactions would be masked by the global buckling behaviour within in a typical analyses. Meanwhile the tip most panel is over constrained relative to the local buckling mode, if this were the primary design driver at this location there would be potential for mass reduction.

Table B.1 shows a comparison of different SE methods for the upper skin cover panels. The local panels had matching stress distributions and similar peak stresses. The modes taken from the baseline represent global wing buckling modes where the panel in question is a part of the response.

Generally the External 2 & 3 step methods are the least accurate as they are subject to truncation errors when transferring ASCII based load and stiffness matrices. The external work applied by each SE model were the same for the External and Part methods, with an average reduction of 15% for the list method. This change is due to an internal calculation of energy relief in the external model due to local displacement.

The LIST method was chosen for the following study as the bulk data between SE and residual structure model are the same, allowing for consistency of design variable specification

TABLE B.1. Super-element Methods Comparison for Upper Skin Cover Panels

	Wing Modes	Panel Buckling Modes			
	baseline	**2 Step**	**3 Step**	**Part**	**List**
Root	0.458	0.677	0.677	0.654	0.653
Mid-span	0.708	1.063	1.063	0.925	0.925
Tip	-	14.334	14.334	14.234	14.262

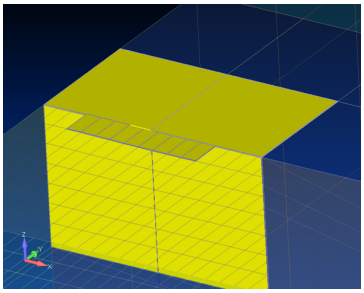
across model boundaries. Common bulk data couples the global variables into the SE variation with the local residual model outputs; without any significant loss in accuracy or additional computation, meaning both coupled and decoupled sensitivities can be extracted. The resulting sensitivity, $\frac{dc_i}{dx_0}$, can then be used to improve the monolithic calculation and as post-optimality sub-gradients in a primal decomposition architecture.

The External 2 & 3 step method was considered with decoupled bulk data allowing for isolated models, however this completely neglects the additional affect of linking variables, skin cover thickness, on the P-K boundary conditions. The external super-element restriction occurs because external super-elements are only known to the code via their structural stiffness, mass, damping, and load matrices. No properties of the outside model may be referenced in the design model, the representative matrices are assumed to be constant with respect to the design model. Additional force-displacement gradients would need to be extracted to chain for linking variable gradients when using the fully decoupled bulk data.

B.2 Multi-Fidelity Panel Modelling

In order to ensure consistency between multi-fidelity models accurate load aggregation and enforcement of constraints is necessary on substructures. Figure B.3(a) shows how element data is transferred from the GFEM to a higher resolution DFEM. Rigid body elements (RBEs) are used to connect elements within the a local area of influence, an expanded model of the boundary can be seen in figure B.3(b). Increasing the panel fidelity on local models ensures a more accurate calculations for stress distribution, and buckling shapes.

(a) Element local Information transition



(b) Expanded RBE Connection

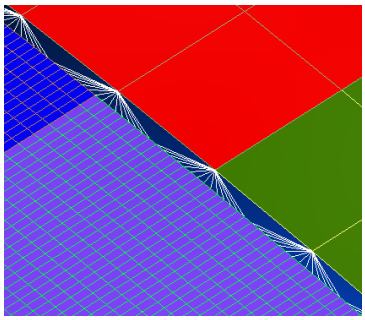


FIGURE B.3. GFEM:DFEM Data Transfer; the coarse mesh is the GFEM and the refined mesh is the DFEM

Table B.2 shows a comparison of the available SE methods acting on a high-fidelity panel at the mid-span. Overall there is a negligible variation in overall panel buckling behaviour between different methods. However the load introductory points along the perimeter of the panel introduce stress concentrations on the edges of the panel, this effect can be better seen in figure B.4. The top panel is the lower resolution stress distribution taken from the GFEM, the bottom panel is the higher resolution version taken from the DFEM. The stress behaviour on the DFEM panel features a much more detailed distribution, whilst the area within 4-5 elements of the interfaces produces significant peaks.

TABLE B.2. Super-element Methods Comparison: High fidelity Panels

	Wing Modes	Panel Buckling Modes			
	baseline	**2 Step**	**3 Step**	**Part**	**List**
Max Stress	2.020E+07	1.264E+08	1.264E+08	1.264E+08	1.264E+08
Ext. Work	-	5.922E+04	5.922E+04	5.846E+03	5.846E+03
Mode 1	0.708	1.041	1.041	0.916	0.916
Mode 2	0.745	1.296	1.296	1.184	1.184
Mode 2	0.820	1.774	1.774	1.674333	1.674333

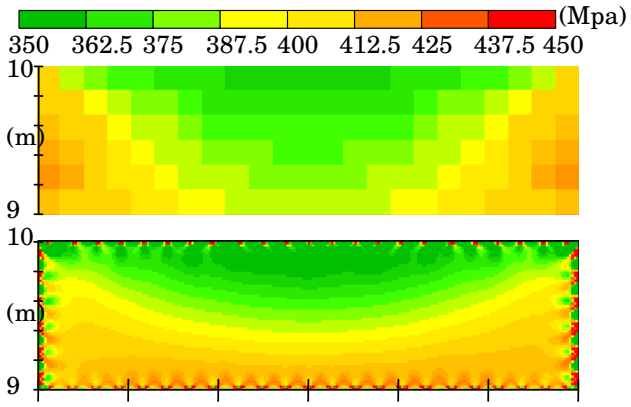


FIGURE B.4. GFEM:DFEM Von Mises Stress Distribution Upper Skin Mid-Panel

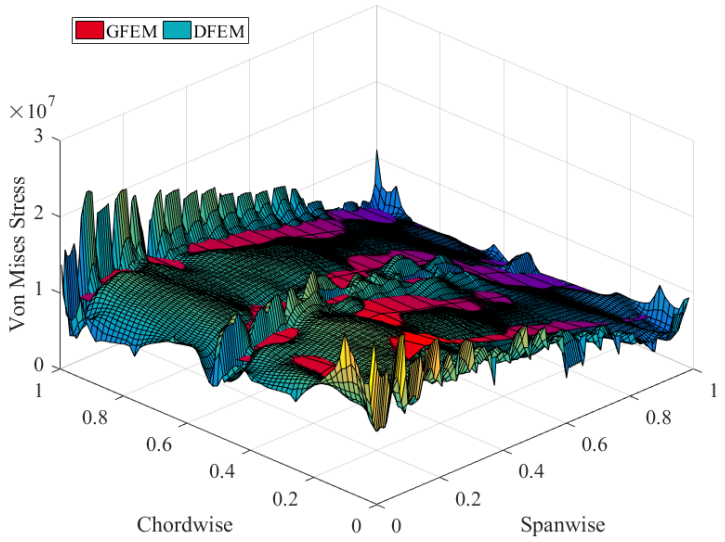


FIGURE B.5. GFEM:DFEM Von-Mises Stress Distribution Upper Skin Root Panel, with additional central spar

Figure B.2 shows a comparison of Von Misses stress distribution across a 1x1 m test panel at the root. The GFEM is represented by the low resolution surface and the DFEM by the higher resolution surface. Overall the DFEM shows a 27% increase in the maximum measured stress and 39% decrease in the average stress, when compared to the GFEM. Expected stress peaks exist at load introductory points along ribs located at 0 and 1m span wise, and spar interfaces at 0, 0.45 and 1m chord wise. As long as data is collected far field of interfaces gradient information for maximum stresses can be comparable. Load dissipation in the DFEM due to stiffeners and greater model flexibility leads to a significantly lower average stress across the panel. The process of substructure load dissipation was isolated and tested for a concurrent parallel architecture using a variable number of stiffeners and then bench-marked against equivalent analytical calculations to ensure robustness across the GFEM and DFEM.

APPENDIX C

APPENDIX C: LOCAL FUNCTION CONSTRAINT SCREENING

This section expands on the mode tracking and constraint aggregation methods used in the local subsystem constraint screening, with additional detail on the implementation and optimisation results from each method.

C.1 Mode Tracking

Tracking eigenpairs between two subsequent analyses in order to ensure the same mode shape is observed throughout an analysis. Tracking is used whenever repeated buckling analyses take place to control discontinuities within the optimisation solution space.

The most accurate methods available use higher-order eigen-pair perturbation algorithms, but can be computationally expensive. The most efficient methods within the literature use cross orthogonality checks, and do not track the mode directly, instead using a statistical indication of the variation in mode shape. [155]

The checks can either use the Modal Assurance Criterion (MAC) or mass triple product. Both are accurate but the MAC is favourable as it eliminates the need for mass orthogonalisation. using only nodal shape displacement, the MAC is bounded 0 to 1 and assigns a correlated set of analyses to measure the likelihood of consistency. The mode shape correlation constant is quantified for a pair for eigen analyses: $\{\psi_A\}$, $\{\psi_B\}$ using the normalised scalar product, shown in equation (C.1).

$$\mathrm{MAC}(r,q) = \frac{\left|\{\psi_A\}_r^T \{\psi_B\}_q^{\otimes}\right|^2}{\left(\{\psi_A\}_r^T . \{\psi_A\}_r^{\otimes}\right)\left(\{\psi_B\}_q^T . \{\psi_B\}_q^{\otimes}\right)} \tag{C.1}$$

Where the subscripts: r, q represent the mode number for analyses A, and B, respectively. The superscript $\otimes$, indicates the complex conjugate. [156, 157]

The resulting matrix for two identical models results in a diagonally dominated correlation, with values of MAC > 0.9 indicating consistent correspondence. The MAC can be seen for the first 25 bucking modes on the upper tip panel in figure C.1. This plot shows the magnitude of correlation between modes shapes using the LIST and External super-element methods, for the same design. The strong diagonal terms indicate that most of the modes are the same. The off-diagonal terms indicate buckling modes which result in similar deflected shapes, but are a different mode number in each analysis.

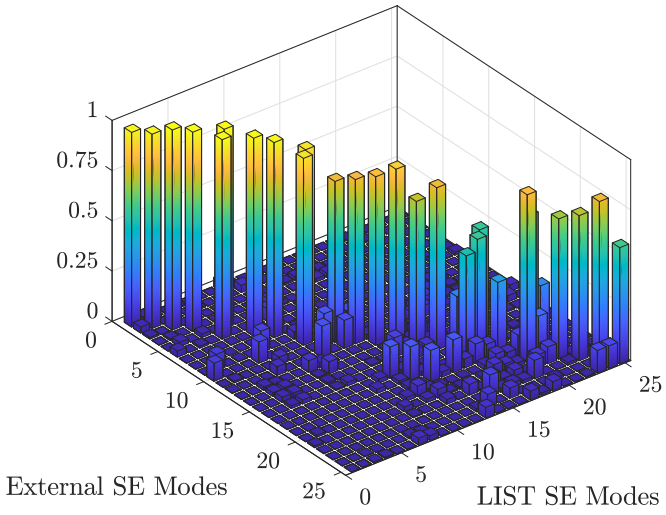


FIGURE C.1. MAC comparison for tip panel with different SE Methods

The MAC is not implemented on the objective or constraint function. Instead the MAC is used to find the desired modes from the new analyses when compared to the previous analyses, using the following process:

Step 1: During the initial analyses the first, 6 of 25 buckling modes are extracted with gradients. Making up the first constraint values $\boldsymbol{\lambda}_i$, and the reference shapes for the subsequent iterations.

Step 2: With each design step the first 25 buckling modes are extracted.

Step 3: The reference shapes from the previous analysis are used to find those with the most like shapes (the greatest MAC) using equation (C.1).

Step 4: The buckling mode and gradients are then extracted for these selected modes, replacing $\boldsymbol{\lambda}_i$.

Step 5: The new modes are then used as the reference shapes for the next analyses.

Using this process it is possible to ensure that the same, or an exceedingly similar, modal behaviour is tracked throughout the optimisation process. Selecting the first 6 of 25 buckling modes should be sufficient to compare the critical buckling cases throughout the analyses. Whilst also allowing for reasonable mode switching and gradual changes in critical mode shapes.

C.2 Constraint Aggregation

Taking a large number of buckling modes at each design step, enough that would be sufficient for the entire spectrum of system behaviour to be captured, then combining them into a single value.

There are many weighted sum and penalty function regimes capable of constraint aggregation, this research used the Kreisselmeier-Steinhauser (KS) function. [158] The KS method was designed for aggregating large-scale gradient-based algorithms, presenting the exact feasible region of the design space and has been widely adopted within buckling analyses problems. The KS function reflexively updates based on the maximum active constraint in a given vector, as shown in equation (C.2).

$$KS_c(\boldsymbol{c_i}(x), \rho_c) = c_{max}(x) + \frac{1}{\rho_c}\left[\ln \sum_{i=1}^{m} e^{\rho_c\left(\boldsymbol{c_i}(x) - c_{max}(x)\right)}\right] \tag{C.2}$$

In equation (C.2) $\boldsymbol{c_i}$ represents a vector of m constraints, c_{max} the maximum constraint, and ρ_c a nominal 'draw-down' aggregation factor, analogous to a penalty factor. Typically a static factor of $\rho_c \approx 50$ produces decent enough results for most problems.

The KS function provides a conservative estimate of the original constraints, with approximation error proportional to ρ_c, and only returns a constraint value $KS_c = c_{max}(x)$, as ρ_c tends to ∞. A static draw-down factor can lead to sub-optimal solutions, especially pronounced at points where multiple constraints are active, at any optima or with excessive amounts of constraints. An adaptive approach to reflexively update the draw-down factor has been applied with respect to a tolerance in the KS design sensitivity with respect to starting design factor, ρ_c. The resulting draw-down factor, ρ_d, is shown in equation (C.3) [61].

$$\log \rho_d = \log\left(\frac{KS'_d}{KS'_c}\right) * \left[\log\left(\frac{KS'_1}{KS'_c}\right)\right]^{-1} * \log\left(\frac{\rho_1}{\rho_c}\right) + \log \rho_c \tag{C.3}$$

In equation (C.3) KS'_c is the design sensitivity, $\frac{dKS}{d\rho_c}$, is a user designated sensitivity tolerance KS'_d, ρ_1 and KS'_1 is the draw-down factor and resulting sensitivity taken some finite step away from the starting factor ρ_c. The enhanced KS function procedure was implemented as follows:

Step 1: Calculate vector of constraints, $\boldsymbol{c_i}$, initialise parameters ρ_c and KS'_d,
Step 2: Calculate KS_c using ρ_c, with the constraint vector using equation (C.2),
Step 3: If $KS'_c < KS'_d$, return KS value; otherwise,
Step 4: Calculate ρ_d using equation (C.3),
Step 5: Calculate KS using ρ_d.

This analyses used starting values $\rho_c = 50$ and $KS'_d = 1e^{-5}$. The first 25 buckling modes were taken at each step and then aggregated with the crippling stress, and local stringer constraints. These responses were concatenated into a single aggregate value, $\bar{c}$.

C.3 Subsystem Optimisation Results

This sample problem used the upper-skin tip panel using the SQP optimisation algorithm, from multiple start points to assess the design space convexity. A preliminary analyses the super-element gradient extraction was assessed in figure C.2 using mode tracking. The iteration histories are presented against the objective mass in kilograms, solutions with only a single iteration could not take a step in any search direction, while those with a star converged to an infeasible solution.

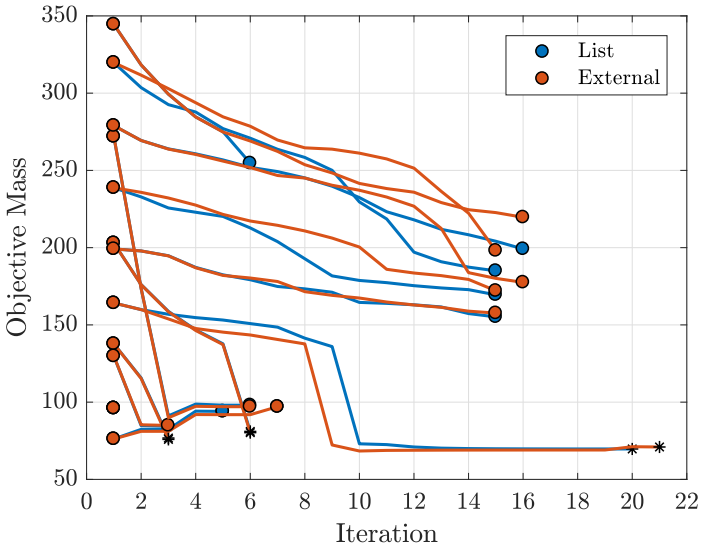


FIGURE C.2. Multi-Start Optimisation with Varying SE Methods for Upper Tip Panel

The optimal solutions with respect to each design point are almost identical, indicating that the gradients extracted between each method are similar. Note the gradients differ from the External methods uses a fixed load along the panel perimeter, whilst the list method accounts for changes due local panel property variation. The changes in loads are not be too large as the test panel is located on the tip of the upper skin.

Mode tracking does not make the design space convex, it simply provides a statistical assurance that similar modal responses are being measured, smoothing the buckling analysis solution space. For more complex designs, or those with a high number of degrees of freedom, like the DFEM panel, the number of potential modes is beyond what could realistically be tracked. Mode tracking is best suited for models with fewer degrees of freedom, or tracking of specific fundamental modal responses. Figure C.3 shows the MAC comparison between the first 25 modes at the starting design y_0 and optimal design y^*. There are no common modes with a MAC > 0.9 between the two design points, indicating a completely different set of mode shapes present.

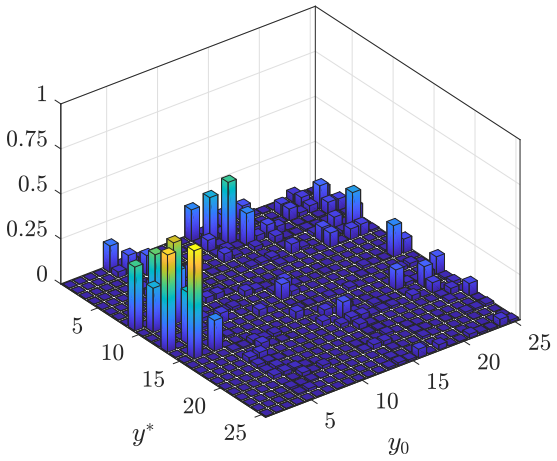


FIGURE C.3. MAC comparison for tip panel between starting and Optimal Designs

Figure C.4 shows the optimisation histories for the upper-skin tip panel, with the MAC and KS based buckling constraint methods from a series of starting points. Objective histories ending with a maximum active constraint greater than zero have a white marker end point. Constraint aggregation performs significantly better than mode tracking methods. Constraint aggregation does not guarantee design convexity, as it simply creates a single representative constraint envelope to conservatively encompass the real multi-constraint problem. The search direction is dependent on the starting position, and as seen in figure C.3, if the starting modal response differs enough from the final solution the search direction may lead to distinctly different converged solutions. In this test 4/10 solutions using constraint aggregation failed to converge to a feasible solution, whilst 9/10 failed when mode tracking was used.

(a) Mode Tracking - 10% Success Rate

(b) Constraint Aggregation - 60% Success Rate

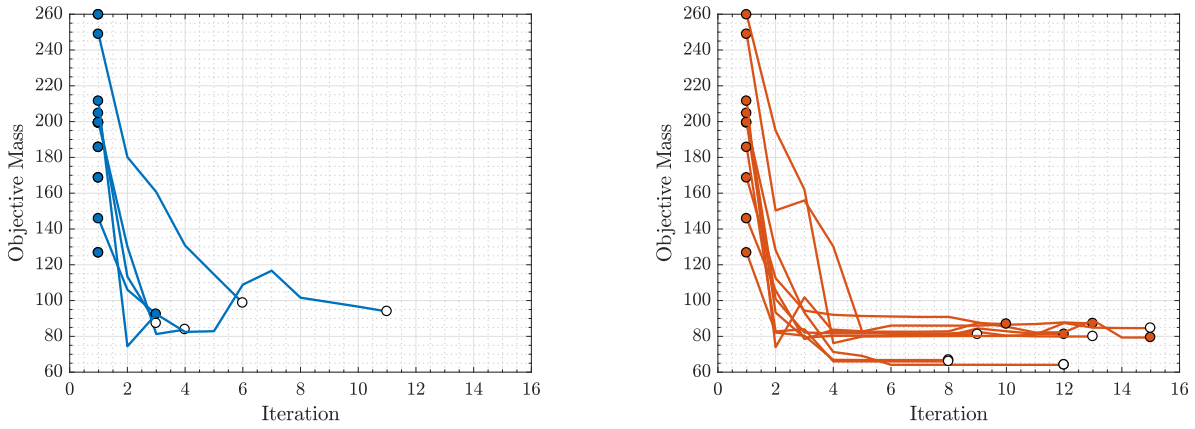


FIGURE C.4. Buckling Constraint Handling Methods

BIBLIOGRAPHY

[1] Roy, R., Hinduja, S., and Teti, R. (2008) Recent advances in engineering design optimisation: Challenges and future trends. *CIRP Annals - Manufacturing Technology,* **57**(2), 697–715.

[2] Gill, P. E., Murray, W., and Wright, M. H. (1981) Practical optimization, Academic press, .

[3] Vanderplaats, G. N. Methods of Mathematical Optimization Vol. 47 of Lecture Notes in Engineering, book section 2, pp. 22–41 Springer Berlin Heidelberg (1989).

[4] Ringertz, U. Large-Scale Structural Design Optimization Vol. 93 of The IMA Volumes in Mathematics and its Applications, book section 10, pp. 235–245 Springer New York (1997).

[5] Rizzi, A. (2011) Modeling and simulating aircraft stability and control—the SimSAC project. *Progress in Aerospace Sciences,* **47**(8), 573–588.

[6] Bhatia, K. G. (2008) A Perspective on Optimization. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA, Reston, VA* Vol. 202008, .

[7] Sobieszczanski-Sobieski, J., James, B. B., and Dovi, A. R. (1985) Structural optimization by multilevel decomposition. *AIAA Journal,* **23**(11), 1775–1782.

[8] Alexandrov, N. and Dennis, J. E. Multilevel algorithms for nonlinear optmization pp. 1–22 Springer (1995).

[9] Massa, A., Oliveri, G., Rocca, P., and Viani, F. (2014) System-by-Design: a new paradigm for handling design complexity. In *Antennas and Propagation (EuCAP), 2014 8th European Conference on* IEEE pp. 1180–1183.

[10] Tosserams, S., Etman, L. F. P., and Rooda, J. E. (2007) An augmented Lagrangian decomposition method for quasi-separable problems in MDO. *Structural and Multidisciplinary Optimization,* **34**(3), 211–227.

[11] ESDU 99019 Constrained multivariate optimisation techniques for the design of aerofoil sections.. `https://www.esdu.com/cgi-bin/ps.pl?sess=unlicensed_1190125155119gyd&t=doc&p=esdu_99019a` (2004) Accessed: 2018-12-15.

[12] Behdinan, K., Perez, R. E., and Liu, H. T. (2005) Multidisciplinary design optimization of aerospace systems. *Proceedings of the Canadian Engineering Education Association (CEEA),* Accessed Online: https://ojs.library.queensu.ca/index.php/PCEEA/article/view/3914.

[13] Tosserams, S., Etman, L. P., and Rooda, J. (2009) A classification of methods for distributed system optimization based on formulation structure. *Structural and Multidisciplinary Optimization,* **39**(5), 503.

[14] Alexandrov, N. M. and Lewis, R. M. (1999) Comparative properties of collaborative optimization and other approaches to MDO. Report, Patent Number: NASA/CR-1999-209354, NAS 1.26:209354, ICASE-99-24.

[15] Martins, J. R. R. A. and Lambe, A. B. (2013) Multidisciplinary Design Optimization: A Survey of Architectures. *AIAA Journal,* **51**(9), 2049–2075.

[16] Martins, P. J. R. R. A. (2014) Large-Scale Multidisciplinary Design Optimization of Aerospace Systems. In Michigan, U. o., (ed.), *SIAM Conference on Optimization*, .

[17] De Jong, K. (2005) Genetic algorithms: a 30 year perspective. *Perspectives on adaptation in natural and artificial systems,* **11**.

[18] Chen, S. Constrained Particle Swarm Optimization - File Exchange - MATLAB Central. `https://uk.mathworks.com/matlabcentral/fileexchange/25986-constrained-particle-swarm-optimizationred` Accessed: 2016-02-23.

[19] Perez, R. l. and Behdinan, K. (2007) Particle swarm approach for structural design optimization. *Computers & Structures,* **85**(19-20), 1579–1588.

[20] Polak, E. (2012) Optimization: algorithms and consistent approximations, Vol. 124, Springer Science & Business Media, .

[21] Shewchuk, J. R. An introduction to the conjugate gradient method without the agonizing pain. `ftp://ftp.unicauca.edu.co/Facultades/.FIET_serepiteencuentasyocupaespacio/DEIC/docs/Materias/` (1994) Accessed: 2016-08-15.

[22] Kennedy, G. J. and Martins, J. (2013) An adjoint-based derivative evaluation method for time-dependent aeroelastic optimization of flexible aircraft. In *Proceedings of the 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Boston, MA*.

[23] Kennedy, G., Kenway, G., and Martins, J. (2014) Towards gradient-based design optimization of flexible transport aircraft with flutter constraints. In *15th AIAA/ISSMO*

Multidisciplinary Analysis and Optimization Conference, Atlanta, Georgia, June pp. 16–20.

[24] Schittkowski, K. (1986) NLPQL: A FORTRAN subroutine solving constrained nonlinear programming problems. *Annals of Operations Research,* **5**(2), 485–500.

[25] MATLAB Optimization Toolbox. (2018) The MathWorks, Natick, MA, USA.

[26] Kaveh, A. and Talatahari, S. (2009) Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. *Computers & Structures,* **87**(5-6), 267–283.

[27] Osaba, E., Yang, X.-S., Diaz, F., Lopez-Garcia, P., and Carballedo, R. (2016) An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. *Engineering Applications of Artificial Intelligence,* **48**, 59–71.

[28] El-Mihoub, T. A., Hopgood, A. A., Nolle, L., and Battersby, A. (2006) Hybrid Genetic Algorithms: A Review. *Engineering Letters,* **13**(2), 14.

[29] Wan, W. and Birch, J. B. (2013) An Improved Genetic Algorithm Using a Directional Search. *Journal of Applied Mathematics,* **2013**, 10.

[30] Jin, Y. (2005) A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing,* **9**(1), 3–12.

[31] Forrester, A. I. J. and Keane, A. J. (2009) Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences,* **45**(1–3), 50–79.

[32] Forrester, A. I., Sóbester, A., and Keane, A. J. (2007) Multi-fidelity optimization via surrogate modelling. In *Proceedings of the Royal Society of London a: mathematical, physical and engineering sciences* The Royal Society Vol. 463, pp. 3251–3269.

[33] Lam, R., Allaire, D. L., and Willcox, K. E. (2015) Multifidelity Optimization using Statistical Surrogate Modeling for Non-Hierarchical Information Sources. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* American Institute of Aeronautics and Astronautics AIAA SciTech.

[34] Haftka, R. T., Villanueva, D., and Chaudhuri, A. (2016) Parallel surrogate-assisted global optimization with expensive functions – a survey. *Structural and Multidisciplinary Optimization,* **54**(1), 3–13.

[35] Voutchkov, I. and Keane, A. Multi-Objective Optimization Using Surrogates Vol. 7 of Adaptation, Learning, and Optimization, book section 7, pp. 155–175 Springer Berlin Heidelberg (2010).

[36] Haftka, R. T., Villanueva, D., and Chaudhuri, A. (2016) Parallel surrogate-assisted global optimization with expensive functions–a survey. *Structural and Multidisciplinary Optimization,* **54**(1), 3–13.

[37] Vankan, W., Maas, R., and Grihon, S. (2014) Efficient optimisation of large aircraft fuselage structures. *The Aeronautical Journal,* **118**(1199), 31–52.

[38] Jones, D. R., Schonlau, M., and Welch, W. J. (Dec, 1998) Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization,* **13**(4), 455–492.

[39] Martí, R., Lozano, J. A., Mendiburu, A., and Hernando, L. (2016) Multi-start methods. *Handbook of Heuristics,* pp. 1–21.

[40] Ugray, Z., Lasdon, L., Plummer, J., Glover, F., Kelly, J., and Martí, R. (2007) Scatter search and local NLP solvers: A multistart framework for global optimization. *INFORMS Journal on Computing,* **19**(3), 328–340.

[41] Lin, Y. and Schrage, L. (2009) The global solver in the LINDO API. *Optimization Methods & Software,* **24**(4-5), 657–668.

[42] Regis, R. G. and Shoemaker, C. A. (2007) A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing,* **19**(4), 497–509.

[43] Lu, J., Han, J., Hu, Y., and Zhang, G. (2016) Multilevel decision-making: A survey. *Information Sciences,* **346–347**, 463–487.

[44] Sinha, A., Malo, P., and Deb, K. (2018) A review on bilevel optimization: from classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation,* **22**(2), 276–295.

[45] Dempe, S. (2018) Bilevel optimization: theory, algorithms and applications. *Fakultät für Mathematik und Informatik,*.

[46] Sobieszczanski-Sobieski, J. and Haftka, R. T. (1997) Multidisciplinary aerospace design optimization: survey of recent developments. *Structural optimization,* **14**(1), 1–23.

[47] Balling, R. J. and Sobieszczanski-Sobieski, J. (1996) Optimization of coupled systems-a critical overview of approaches. *AIAA journal,* **34**(1), 6–17.

[48] Dantzig, G. B. and Wolfe, P. (1960) Decomposition principle for linear programs. *Operations research,* **8**(1), 101–111.

[49] Yoshimura, M. and Izui, K. (2002) Smart optimization of machine systems using hierarchical genotype representations. *Journal of Mechanical Design,* **124**(3), 375–384.

[50] El-Beltagy, M. A. and Keane, A. J. (1999) A comparison of various optimization algorithms on a multilevel problem. *Engineering Applications of Artificial Intelligence,* **12**(5), 639–654.

[51] Mendonca, M. W. Multilevel Optimization: Convergence Theory, Algorithms and Application to Derivative-Free Optimization Thesis (2009).

[52] Stockmeyer, L. J. (1976) The polynomial-time hierarchy. *Theoretical Computer Science,* **3**(1), 1–22.

[53] Sobieszczanski-Sobieski, J. (2008) Integrated system-of-systems synthesis. *AIAA journal,* **46**(5), 1072–1080.

[54] Stackelberg, H. (1952) The theory of the market economy, Oxford University Press, .

[55] Downey, R. G. and Fellows, M. R. (2012) Parameterized complexity, Springer Science & Business Media, .

[56] El-Beltagy, M. A. and Keane, A. J. (1998) Optimization for Multilevel Problems: A Comparison of Various Algorithms. *Adaptive Computing in Design and Manufacture,* pp. 111–120.

[57] El-Beltagy, M. A., Nair, P. B., and Keane, A. J. (1999) Metamodeling techniques for evolutionary optimization of computationally expensive problems: Promises and limitations. *Gecco-99: Proceedings of the Genetic and Evolutionary Computation Conference,* pp. 196–203.

[58] Boyd, S., Xiao, L., Mutapcic, A., and Mattingley, J. (2007) Notes on decomposition methods. *Notes for EE364B, Stanford University,* pp. 1–36.

[59] Clark, P. A. and Westerberg, A. W. (1983) Optimization for design problems having more than one objective. *Computers & Chemical Engineering,* **7**(4), 259–278.

[60] Haftka, R. T. (1985) Simultaneous analysis and design. *AIAA journal,* **23**(7), 1099–1103.

[61] Martins, J. and Poon, N. M. (2005) On structural optimization using constraint aggregation. In *VI World Congress on Structural and Multidisciplinary Optimization WCSMO6, Rio de Janeiro, Brasil*.

[62] Kaveh, A. and Ghazaan, M. I. (2014) Enhanced colliding bodies optimization for design problems with continuous and discrete variables. *Advances in Engineering Software,* **77**, 66–75.

[63] Nash, S. G. (2000) A multigrid approach to discretized optimization problems. *Optimization Methods and Software,* **14**(1-2), 99–116.

[64] Braun, R. D. (1996) Collaborative optimization: an architecture for large-scale distributed design.

[65] Sobieszczanski-Sobieski, J., Altus, T. D., Phillips, M., and Sandusky, R. (2003) Bilevel integrated system synthesis for concurrent and distributed processing. *AIAA journal,* **41**(10), 1996–2003.

[66] Kim, H. M., Michelena, N. F., Papalambros, P. Y., and Jiang, T. (2003) Target Cascading in Optimal System Design. *Journal of Mechanical Design,* **125**(3), 474–480.

[67] Roth, B. and Kroo, I. (2008) Enhanced collaborative optimization: application to an analytic test problem and aircraft design. In *12th AIAA/ISSMO multidisciplinary analysis and optimization conference* p. 5841.

[68] Aliaga-Aguilar, H. and Cuerno-Rejado, C. (2018) Generic parameter penalty architecture. *Structural and Multidisciplinary Optimization,* pp. 1–11.

[69] Balling, R. and Sobieszczanski-Sobieski, J. (1994) An algorithm for solving the system-level problem in multilevel optimization. In *5th Symposium on Multidisciplinary Analysis and Optimization* p. 4333.

[70] Blouin, V. Y., Lassiter, J. B., Wiecek, M. M., and Fadel, G. M. (2005) Augmented Lagrangian coordination for decomposed design problems. In *Proceedings of the 6th world congress on structural and multidisciplinary optimization*.

[71] Miguel, A.-V. D. (2001) Two decomposition algorithms for nonconvex optimization problems with global variables, Stanford University, .

[72] Miguel, A.-V. D. and Murray, W. (2006) A local convergence analysis of bilevel decomposition algorithms. *Optimization and Engineering,* **7**(2), 99–133.

[73] Chittick, I. R. and Martins, J. R. (2009) An asymmetric suboptimization approach to aerostructural optimization. *Optimization and Engineering,* **10**(1), 133–152.

[74] Kennedy, G., Martins, J., and Hansen, J. (2008) Aerostructural optimization of aircraft structures using asymmetric subspace optimization. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* p. 5847.

[75] Benders, J. F. (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik,* **4**(1), 238–252.

[76] Lootsma, F. and Ragsdell, K. (1988) State-of-the-art in parallel nonlinear programming. *Parallel Comput,* **6**(2), 133–155.

[77] Wang, J., Xie, F., Zheng, Y., Zhang, J., Yang, B., and Ji, T. (2018) Virtual Stackelberg game coupled with the adjoint method for aerodynamic shape optimization. *Engineering Optimization,* **50**(10), 1733–1754.

[78] Park, H., Michelena, N., Kulkarni, D., and Papalambros, P. (2001) Convergence criteria for hierarchical overlapping coordination of linearly constrained convex design problems. *Computational Optimization and Applications,* **18**(3), 273–293.

[79] Ollar, J., Toropov, V., and Jones, R. (2017) Sub-space approximations for MDO problems with disparate disciplinary variable dependence. *Structural and Multidisciplinary Optimization,* **55**(1), 279–288.

[80] Haftka, R. T. and Watson, L. T. (2006) Decomposition theory for multidisciplinary design optimization problems with mixed integer quasiseparable subsystems. *Optimization and Engineering,* **7**(2), 135–149.

[81] Sobieszczanski-Sobieski, J. (1989) Optimization by decomposition: a step from hierarchic to non-hierarchic systems.

[82] Bloebaum, C. L., Hajela, P., and Sobieszczanski-Sobieski, J. (1992) Non-hierarchic system decomposition in structural optimization. *Engineering Optimization+ A35,* **19**(3), 171–186.

[83] Sobieszczanski-Sobieski, J., Agte, J. S., and Sandusky, R. R. (2000) Bilevel integrated system synthesis. *AIAA journal,* **38**(1), 164–172.

[84] Shin, M.-K. and Park, G.-J. (2005) Multidisciplinary design optimization based on independent subspaces. *International Journal for Numerical Methods in Engineering,* **64**(5), 599–617.

[85] Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning,* **3**(1), 1–122.

[86] Arati Singh, P. R. Augmented Lagrangian & the Method of Multipliers. (September, 2017) http://www.cs.cmu.edu/ pradeepr/convexopt/Lecture_Slides/.

[87] Pearl, J. (1984) Heuristics: intelligent search strategies for computer problem solving, Addison-Wesley Pub. Co., Inc., Reading, MA, .

[88] Liu, X., Jiang, X., and Li, S. (2018) Optimal Landing Site Selection for Planetary Landing Using Multilevel Optimization. In *4th IAA Conference on Dynamics and Control of Space Systems*.

[89] Schuster, A., Scherer, J., Führer, T., Bach, T., and Kohlgrüber, D. (2016) Automated Sizing Process of a Complete Aircraft Structure for the Usage within a MDO Process, Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV, Bonn, Document No: 420137.

[90] Sobieszczanski-Sobieski, J. and Haftka, R. T. (1997) Multidisciplinary aerospace design optimization: survey of recent developments. *Structural optimization,* **14**(1), 1–23.

[91] T Ralphs, A Hassanzadeh, J. W. M. G. M. G. S. Decomposition Methods for Discrete Optimization. `https://coral.ie.lehigh.edu/~ted/files/talks/DECOMP-ICS13.pdf` (January, 2013) Accessed: 2015-12-02.

[92] Kroo, I., Altus, S., Braun, R., Gage, P., and Sobieski, I. (1994) Multidisciplinary optimization methods for aircraft preliminary design. In *5th symposium on multidisciplinary analysis and optimization* p. 4325.

[93] Zadeh, P., Toropov, V., and Wood, A. Collaborative optimization framework based on the interaction of low-and high-fidelity models and the moving least squares method. In *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 14th AIAA/ASME/AHS Adaptive Structures Conference 7th* p. 1711.

[94] Balling, R. and Rawlings, M. (2000) Collaborative optimization with disciplinary conceptual design. *Structural and Multidisciplinary Optimization,* **20**(3), 232–241.

[95] Tosserams, S., Etman, L., Papalambros, P., and Rooda, J. (2006) An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers. *Structural and multidisciplinary optimization,* **31**(3), 176–189.

[96] Allison, J., Walsh, D., Kokkolaras, M., Papalambros, P. Y., and Cartmell, M. (2006) Analytical target cascading in aircraft design. In *Proceedings of the 44th AIAA aerospace sciences meeting and exhibit* Vol. AIAA-2006-1326, pp. 9–12.

[97] Cooper, A. B., Georgiopoulos, P., Kim, H. M., and Papalambros, P. Y. (2005) Analytical Target Setting: An Enterprise Context in Optimal Product Design. *Journal of Mechanical Design,* **128**(1), 4–13.

[98] Kim, H. M., Rideout, D. G., Papalambros, P. Y., and Stein, J. L. (2003) Analytical Target Cascading in Automotive Vehicle Design. *Journal of Mechanical Design,* **125**(3), 481–489.

[99] Allison, J., Roth, B., Kokkolaras, M., Kroo, I., and Papalambros, P. Y. (2006) Aircraft family design using decomposition-based methods. In *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*.

[100] Gardenghi, M., Wiecek, M. M., and Wang, W. (2013) Biobjective optimization for analytical target cascading: optimality vs. achievability. *Structural and Multidisciplinary Optimization,* **47**(1), 111–133.

[101] Zhang, Y., Zhang, G., Qu, T., Liu, Y., and Zhong, R. Y. (2017) Analytical target cascading for optimal configuration of cloud manufacturing services. *Journal of Cleaner Production,* **151**, 330–343.

[102] Bindolino, G., Ghiringhelli, G., and Ricci, S. (2016) A Multi-Level Procedure For The Preliminary Sizing Of Aircraft Wing-Box Structures. *Aerotecnica Missili and Spazio,* **84**(4), 164–181.

[103] Ciampa, P. D., Nagel, B., and Van Tooren, M. (2010) Global local structural optimization of transportation aircraft wings. In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Orlando, USA, 12-15 April 2010; AIAA 2010-3098* American Institute of Aeronautics and Astronautics (AIAA).

[104] Li, Y., Lu, Z., and Michalek, J. J. (2008) Diagonal quadratic approximation for parallelization of analytical target cascading. *Journal of Mechanical Design,* **130**(5), 051402.

[105] Roth, B. D. and Kroo, I. M. (2008) Enhanced collaborative optimization: a decomposition-based method for multidisciplinary design. In *ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* American Society of Mechanical Engineers pp. 927–936.

[106] Allison, J., Roth, B., Kokkolaras, M., Kroo, I., and Papalambros, P. (2006) Aircraft family design using decomposition-based methods. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* p. 6950.

[107] Kim, H. M., Chen, W., and Wiecek, M. M. (2006) Lagrangian coordination for enhancing the convergence of analytical target cascading. *AIAA journal,* **44**(10), 2197–2207.

[108] Michelena, N., Park, H., and Papalambros, P. Y. (2003) Convergence Properties of Analytical Target Cascading. *AIAA Journal,* **41**(5), 897–905.

[109] Zhang, D., Song, B., and Wang, P. (2017) A parallel bi-level multidisciplinary design optimization architecture with convergence proof for general problem. *Engineering Optimization,* **49**(4), 654–674.

[110] Tosserams, S., Etman, L., and Rooda, J. (2008) Augmented Lagrangian coordination for distributed optimal design in MDO. *International journal for numerical methods in engineering,* **73**(13), 1885–1910.

[111] Ellman, T., Keane, J., Schwabacher, M., and Yao, K.-T. (1997) Multilevel modelling for engineering design optimization. *AI EDAM,* **11**(5), 357–378.

[112] Avraamidou, S. and Pistikopoulos, E. N. (2018) B-POP: Bi-level Parametric Optimization Toolbox. *Computers & Chemical Engineering,*.

[113] Haftka, R. T., Sobieszczanski-Sobieski, J., and Padula, S. L. (1992) On options for interdisciplinary analysis and design optimization. *Structural optimization,* **4**(2), 65–74.

[114] Faísca, N. P., Dua, V., Rustem, B., Saraiva, P. M., and Pistikopoulos, E. N. (2007) Parametric global optimisation for bilevel programming. *Journal of Global Optimization,* **38**(4), 609–623.

[115] Liu, Q., Jrad, M., Mulani, S. B., and Kapania, R. K. (2016) Global/Local Optimization of Aircraft Wing Using Parallel Processing. *AIAA Journal,* **54**(11), 3338–3348.

[116] Dezyani, M., Dalayeli, H., Yousefi, S., and Farrokhfal, H. (2017) Two-Level Design Optimization for Compression Stiffened Panel. *Iranian Journal of Science and Technology, Transactions of Mechanical Engineering,*.

[117] Grihon, S., Krog, L., and Bassir, D. (2009) Numerical Optimization applied to structure sizing at AIRBUS: A multi-step process. *International Journal for Simulation and Multidisciplinary Design Optimization,* **3**(4), 432–442.

[118] Chintapalli, S., Elsayed, M. S. A., Sedaghati, R., and Abdo, M. (2010) The development of a preliminary structural design optimization method of an aircraft wing-box skin-stringer panels. *Aerospace Science and Technology,* **14**(3), 188–198.

[119] Ragon, S. A., Gürdal, Haftka, R. T., and Tzong, T. J. (2003) Bilevel Design of a Wing Structure Using Response Surfaces. *Journal of Aircraft,* **40**(5), 985–992.

[120] Mulani, S. B., Slemp, W. C., and Kapania, R. K. (2013) EBF3PanelOpt: an optimization framework for curvilinear blade-stiffened panels. *Thin-Walled Structures,* **63**, 13–26.

[121] Giesen, J. and Laue, S. Distributed convex optimization with many convex constraints. (2016).

[122] Mallozzi, L., Reina, G. P., Russo, S., and de Nicola, C. (2016) Game Theoretical Tools for Wing Design. In *International Workshop on Machine Learning, Optimization and Big Data* Springer pp. 419–426.

[123] Nishihara, R., Lessard, L., Recht, B., Packard, A., and Jordan, M. I. A general analysis of the convergence of ADMM. (2015).

[124] Xiao, A., Zeng, S., Allen, J. K., Rosen, D. W., and Mistree, F. (2005) Collaborative multidisciplinary decision making using game theory and design capability indices. *Research in Engineering Design,* **16**(1-2), 57–72.

[125] Büyüközkan, G. and Arsenyan, J. (2012) Collaborative product development: a literature overview. *Production Planning & Control,* **23**(1), 47–66.

[126] Lewis, K. and Mistree, F. (1997) Modeling interactions in multidisciplinary design: A game theoretic approach. *AIAA journal,* **35**(8), 1387–1392.

[127] Moerland, E., Zill, T., Nagel, B., Spangenberg, H., Schumann, H., and Zamov, P. (2012) Application of a distributed MDAO framework to the design of a short-to medium-range aircraft, Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV, .

[128] Cavagna, L., Ricci, S., and Travaglini, L. (2011) NeoCASS: an integrated tool for structural sizing, aeroelastic analysis and MDO at conceptual design level. *Progress in Aerospace Sciences,* **47**(8), 621–635.

[129] Van der Velden, A. (2010) Isight design optimization methodologies. *ASM handbook,* **22**, 79–101.

[130] August, T. N. and Alan, W. W. (2016) Bi-Level Optimization of a Conceptual Metallic Wing Box with Stiffness Constraints. In *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* American Institute of Aeronautics and Astronautics AIAA SciTech.

[131] Lisandrin, P. and Tooren, M. (2007) High-order finite elements reduced models for modal analysis of stiffened panels. *International Journal of Mechanics and Materials in Design,* **3**(2), 111–127.

[132] MSC-Software MSC Nastran 2012 Quick Reference Guide. `https://books.google.co.uk/books?id=lTUrKh_iTSEC` (2011) Accessed: 2015-06-01.

[133] MatWeb-LLC Overview of materials for 2000 Series Aluminum Alloy. `http://www.matweb.com/search/datasheet_print.aspx?matguid=2076184469d740af9f86b0d69b2e42ff` (2016) Accessed: 2016-11-22.

[134] Bruhn, E. F., Bollard, R., et al. (1973) Analysis and design of flight vehicle structures, Jacobs Publishing Indianapolis, .

[135] Azar, J. J. and Peery, D. J. Aircraft structures. (1982).

[136] Megson, T. H. G. Aircraft structures for engineering students. (2013).

[137] Lambe, A. B. and Martins, J. R. (2012) Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization,* **46**(2), 273–284.

[138] Xu, M., Fadel, G., Wiecek, M. M., and Guarneri, P. (2015) An efficient parallel coordination method for decomposition-based optimization using two duality theorems. In *11th World Congress on Structural and Multidisciplinary Optimisation*.

[139] Müller, J. User guide for modularized surrogate model toolbox. `https://uk.mathworks.com/matlabcentral/fileexchange/38530-surrogate-model-optimization-toolbox` (2012) Accessed: 2017-08-10.

[140] Kim, T. S. and Kim, Y. Y. (2000) Mac-based mode-tracking in structural topology optimization. *Computers & Structures,* **74**(3), 375–383.

[141] Dunning, P. D., Stanford, B., and Kim, H. A. (2015) Level-set topology optimization with aeroelastic constraints. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* p. 1128.

[142] de Baar, J. H., Dwight, R. P., and Bijl, H. (2014) Improvements to gradient-enhanced Kriging using a Bayesian interpretation. *International Journal for Uncertainty Quantification,* **4**(3).

[143] Ollar, J., Mortished, C., Jones, R., Sienz, J., and Toropov, V. (2017) Gradient based hyperparameter optimisation for well conditioned kriging metamodels. *Structural and Multidisciplinary Optimization,* **55**(6), 2029–2044.

[144] Allison, J. T. and Papalambros, P. Y. (2010) Consistency constraint allocation in augmented lagrangian coordination. *Journal of Mechanical Design,* **132**(7), 071007.

[145] Horn, R. A. (1990) The Hadamard Product. In *Proc. Symp. Appl. Math* Vol. 40, pp. 87–169.

[146] Luenberger, D. G., Ye, Y., et al. Linear and nonlinear programming. (1984).

[147] Michalek, J. J. and Papalambros, P. Y. (2004) An efficient weighting update method to achieve acceptable consistency deviation in analytical target cascading. In *ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* American Society of Mechanical Engineers pp. 159–168.

[148] Loftsgaarden, D. O., Quesenberry, C. P., et al. (1965) A nonparametric estimate of a multivariate density function. *The Annals of Mathematical Statistics,* **36**(3), 1049–1051.

[149] Simpson, T. W., Rosen, D., Allen, J. K., and Mistree, F. (1998) Metrics for assessing design freedom and information certainty in the early stages of design. *Journal of Mechanical Design,* **120**(4), 628–635.

[150] Chen, W., Simpson, T. W., Allen, J. K., and Mistree, F. (1999) Satisfying ranged sets of design requirements using design capability indices as metrics. *Engineering Optimization,* **31**(5), 615–619.

[151] Svanberg, K. (2007) MMA and GCMMA – versions September 2007. *Optimization and Systems Theory,* **104**.

[152] Reenskaug, T. (2003) The model-view-controller (MVC) its past and present. *University of Oslo Draft,*.

[153] Grobler, C. Multi-Objective Parallelization of Efficient Global Optimization , thesis, University of Pretoria (2016).

[154] Hwang, J. and Martins, J. (2012) GeoMACH: geometry-centric MDAO of aircraft configurations with high fidelity. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* p. 5605.

[155] Eldred, M., Venkayya, V., and Anderson, W. (1995) Mode tracking issues in structural optimization. *AIAA journal,* **33**(10), 1926–1933.

[156] Pastor, M., Binda, M., and Harčarik, T. (2012) Modal assurance criterion. *Procedia Engineering,* **48**, 543–548.

[157] Allemang, R. J. (2003) The modal assurance criterion – twenty years of use and abuse. *Sound and vibration,* **37**(8), 14–23.

[158] Kreisselmeier, G. and Steinhauser, R. (1980) Systematic control design by optimizing a vector performance index. In *Computer Aided Design of Control Systems* pp. 113–117 Elsevier.